MONTE CARLO APPROACH TO DYNAMIC PSA: NEURAL SOLUTION OF EQUATIONS DESCRIBING CORE TRANSIENTS

Marzio Marseguerra

Enrico Padovani

Marco E. Ricotti

Department of Nuclear Engineering
Polytechnic of Milan
Milan
Italy

ABSTRACT

The PSA analysis of a real plant represent a formidable computational task usually afforded either with an analytical approach based on the theory of the Markov chains or with a Monte Carlo simulation. In our opinion this latter methodology, thanks to its unique flexibility features, represents the only viable approach to the problem when time dependencies have impacts on the analysis: examples are time dependent transition rates (ageing), timing of the protection, control and safety systems, operator actions etc.

Moreover the PSA analysis of a real plant demands taking into account the process variable dynamics when the evolution of the underlying physical process interacts with the system hardware configuration, e.g. when the process variables influence the failure rates or activate the protection systems. The inclusion of these dynamic aspects dramatically burdens the analysis: a solution could be presently attempted only through short cuts to the solution of the deterministic equations governing the evolution of the process variables.

In the present paper we consider the application of a multilayered neural network for the solution of the mathematical models related to the core behaviour of a PWR under varying thermal-hydraulic conditions. Since the neural network works very rapidly, this approach seems to be a good candidate for being included in a Monte Carlo dynamic PSA code which requires solving thousands of times the model equations relating to the different hardware plant configurations. Possible approximations thereby introduced could be tolerated if comparable with those following from the uncertainties of the stochastic parameters. The time reduction advantage is expected to increase when the future parallel computers become widely available.

1. INTRODUCTION

Monte Carlo simulation methods (Kamat and Riley, 1975), (Kamat and Franzmeier, 1976), (Gelbard and Spanier, 1969), (Kalos and Whitlock, 1986), (Kumamoto et al., 1977), (Lewis and Bohm, 1984), (Lewis and Zhuguo, 1986) have been extensively used in probabilistic safety assessment (PSA) studies especially when dealing with systems which are too complex and too large to be managed by analytical methods such as those derived from the theory of the Markov chains which gives rise to a huge number of differential equations. However it is important to note that, in order to be close to reality, one should also take into account that the underlying system dynamics might influence the PSA of a real plant. For example, when a plant component makes a transition, such as a stochastic failure or a deterministic transition demanded by the control/protection system, variations in the process variables evolution are also originated. If these do not appear or if their effect is negligible, than the classic, static PSA based on the event tree/fault tree methodology (U.S. Nuclear Regulatory Commission, 1981) leads to satisfactory results. On the contrary, if a component transition is strongly sensitive to the process variables values, such as a pump cavitation induced by a decrease of the coolant pressure due to a pressurizer failure, then any reliability analysis should properly take into account these dynamic aspects and therefore a dynamic approach, namely a dynamic PSA, is almost mandatory (Devooght and Smidts, 1992a), (Devooght

and Smidts, 1992b), (Siu and Deoss, 1989), (Siu, 1994), (Marseguerra and Zio, 1992), (Marseguerra and Zio, 1993). Unfortunately this approach enormously increases the complexity of the computer programs and the burden of the calculations. Indeed a real plant is likely to have a large number of possible hardware configurations and for each of them a mathematical model describing the physical evolution of the process variables must be introduced and managed. Moreover the solution of the equations pertaining to these models must be carried out with time steps related to the time constants of the underlying process, which are orders of magnitude smaller than the hardware stochastic transition times. The above remarks suggest that, with the present computers, a dynamic PSA of a real plant is almost prohibitive unless suitably semplified models describing the evolution of the physical processes in all the system configurations and efficient algorithms for their solutions are developed.

In the present paper we investigate the feasibility of solving the equations relating to core transients by resorting to the artificial neural network (ANN) methodology.

The ANNs represent one of the fastest developing methodologies which has been extensively applied in the past decade to quite different areas such as pattern recognition, data compression and dynamic control (Rumelhart and McClelland, 1986). All these applications exploit the connectionist feature of the ANNs which allows them to recognize in real time complex patterns even when the information they receive is incomplete or corrupted by noise. In the nuclear field the inclusion of the ANNs into the protection control system may be beneficial in terms of forecasting in real time of process variables and therefore they may be used to estimate plant parameters and as a diagnostic tool. A review of the potential applications of the ANN methodology to the operation of nuclear reactor plants has been given by Uhrig (1991); examples of specific applications concern the prediction of important performance parameters (Roh et al., 1991), (Parks et al., 1991), (Guo and Uhrig, 1992), (Kim et al., 1993), the failure detection (Ciftcioglu et al., 1991), (Cheon et al., 1993), (Marseguerra and Zio, 1994a), (Marseguerra and Zio, 1994b), the safety control (Jouse and Williams, 1992), (Jouse and Williams, 1993).

In the present application, we firstly set up a suitable reactor core model. Then, by considering a rather wide set of transients lasting time intervals of the order of a quarter of an hour, we investigate whether it is possible to use the process variables, computed with the selected core model during these transients, for suitably designing and training an ANN. The procedure is successful if the ANN, after training completion, is capable of predicting with reasonable accuracy the values of the process variables of interest in correspondence of new, never seen before, test transients.

In Section 2, we outline the main features of the multilayered, supervised artificial neural network and the error back-propagation algorithm. In Section 3, we sketch the physical model of the reactor core and discuss the associated mathematical equations. In Section 4 we present the approach employed for the solution of the problem and in Section 5 the results obtained. The paper then closes with some concluding remarks in Section 6.

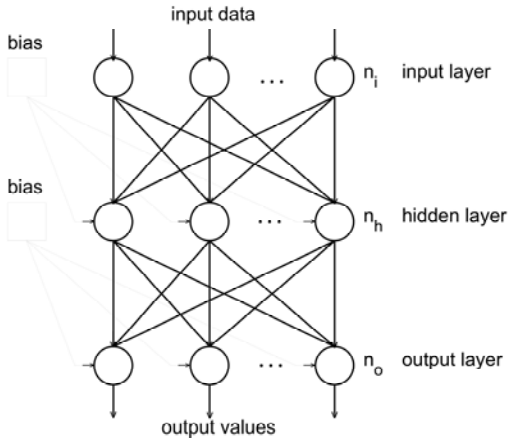


Fig. 1 - Scheme of a three-layered, feedforward neural network.

2. THE SUPERVISED, MULTILAYERED, FEEDFORWARD NEURAL NETWORK AND THE ERROR BACK-PROPAGATION ALGORITHM

Neural networks are information processing systems composed of simple processing elements (nodes) linked by weighted connections (Rumelhart and McClelland, 1986), (Cichocki and Unbehauen, 1993), (Sanchez-Sinencio and Lau, 1992). In its simplest form, the multilayered feedforward neural network consists of three layers of processing elements: the input, the hidden and the output layers, with n_i, n_h and n_o nodes respectively (Fig. 1). This choice comes from the fact that almost any non-linear in-out mapping may be achieved with such a topology (White, 1989). The nodes of a layer are only connected with those of the following layer, the strength of the connection between nodes *i* and *j* being w_{ij}. The important matter of choosing the number of nodes in each layer is still an open question, heavily depending on the specific problem. If the number of nodes is too small, the network does not learn the problem; if it is too large the network is not able to generalize after training completion. In lack of well-established rules, we followed a parsimony criterion, trying to keep lower the number of nodes in each layer. The signal is processed forwardly from the input to the output layer. Each node collects the output values, multiplied by the connection weights, from all the nodes of the preceding layer, process this information through a sigmoidal function $f(x) = (1 + e^{-x})^{-1}$ and then delivers the result of all the connections towards the nodes of the successive layer. In the present work both input and hidden layers have the additional bias node, which may be seen as a threshold in the argument of the activation function and whose output always equals unity.

The values of the connection weights are determined through a training procedure. In this case we have adopted the usual error back-propagation algorithm which follows from the general gradient descent method (Muller and reinhardt, 1991). In short, the algorithm consists in repeatedly presenting a set of n_p input and corresponding output patterns, and iteratively adjusting the values of the connection weights so to minimize the *average squared output deviation error function*, or *Energy function*, over all training patterns and all n_o outputs, defined as

$$D = \frac{1}{2\, n_p\, n_o} \sum_{p=1}^{n_p} \sum_{l=1}^{n_o} (t_{pl} - o_{pl})^2$$

where, t_{pl} and o_{pl} are the true and the network-computed values of the l-th output node, to the p-th pattern presented. The back-propagation algorithm perform the steepest descent on a surface in a weight space, whose height at any point is equal to the error function. To be more specific, at the n-th iteration the weight w_{ij} of each connection is updated according to the following relaxation procedure

$$\Delta w_{ij}^{(n)} = -\alpha \frac{\partial D}{\partial w_{ij}} + \eta\ \Delta w_{ij}^{(n-1)}$$

where α and η are the learning and the momentum coefficients. The first term in the rhs, called gradient term, follows from the usual method of steepest descent. The second term, called momentum term, introduces a sort of hysteresis effect: in the weight space, where the error surface D is strongly curved, the gradient term is large and oscillations in the w_{ij} values may result. The momentum term adds a memory so that abrupt changes of the weights are avoided or at least reduced. Through this training procedure, the network is able to build an internal representation of the input/output mapping of the problem under investigation. The success of the training strongly depends on the normalization of the data and on the choice of the training parameters, namely the learning and the momentum coefficients. Unfortunately there are not general expression for selecting the optimal α and η parameters and one has to resort to his experience. In this work, each signal has been transferred within the interval (0.2, 0.8) by an affine mapping; the learning coefficient and the momentum factor have been selected in the range (0.6, 0.8); moreover the connection weights have been initialized randomly within the interval (-0.3, 0.3).

After the training is completed, the final connection weights are kept fixed and new input patterns are presented to the network which is capable of recalling the information, stored in the connection weights during training, to produce the corresponding output, coherent with the internal representation of the input/output mapping. Notice that the nonlinearity of the sigmoidal function of the processing elements allows the neural network to learn arbitrary nonlinear mapping. Moreover, each node acts independently of all the others and its functioning relies only on the local information provided through the adjoining connections. In other words, the functioning of one node does not depend on the states of those other nodes to which it is not connected. This allows for efficient distributed representation and parallel processing, and for an intrinsic fault-tolerance and generalization capability.

These attributes render the artificial neural networks a powerful tool for signal processing, nonlinear mappings and near-optimal solution to combinatorial optimization problems.

3. THE CORE MODEL

The core model here considered is part of the TRAP code (Ricotti, 1994), developed in our Department, which treats a PWR reactor of the kind AP-600 Westinghouse. The core dynamics is modeled by a system of 13 partial differential equations and 8 correlations which take into account the following aspects (meanings of symbols reported in the Nomenclature at the end of the paper):

3.1 Neutron Dynamics

We adopted the usual one-velocity, point-reactor model with six groups of delayed neutrons modeled by 7 differential equations in the neutron population $n(t)$ and in the precursor concentration $c_j(t)$, (j=1, 6). The feedback effects are assumed to act independently one another, so that the reactivity coefficient appearing in the neutron balance equation may be written as

$$\rho = \rho_{T_{f,eff}} + \rho_{\bar{T}_m} + \rho_{\bar{\gamma}_m} + \rho_{CR} \qquad (1)$$

where the various terms represent the contributes due to an effective fuel temperature $T_{f,eff}$ (Doppler effect), to a mean moderator temperature $\bar{T}_m$, to a mean moderator density $\bar{\gamma}_m$ and to the control rod positions CR, respectively. Each term in equation (1) is expressed as the product of a known feedback coefficient times the difference between the current and the reference value (i.e. at full power) of the corresponding variable: for example, for the Doppler term we have:

$$\rho_{T_{f,eff}}(t) = A_{T_{f,eff}} \left(T_{f,eff}(t) - \tilde{T}_{f,eff} \right) \qquad (2)$$

The average values $\bar{T}_m$ and $\bar{\gamma}_m$ have been obtained from the thermalhydraulics calculations detailed below, with reference to a core consisting of three parallel, independent channels which, according to a sinusoidal neutron flux form factor, deliver 47.7%, 33.3% and 19% of the total power. The average values of T_m and γ_m along each channel, further averaged over the relative powers of the three channels, give rise to the values to be used in eq. (2). At each time t, the temperature $T_{f,eff}$ is given by a linear combination of the temperature $T_{p,r=0}$ of the pellet centre and of the outer cladding temperature $T_{c,r=Roc}$

$$T_{f,eff} = 4/9\ T_{p,r=0} + 5/9\ T_{c,r=Roc} \qquad (3)$$

The computation of T_p and T_c is illustrated in the next subsection.

3.2 Fuel to coolant heat transfer

The time dependent Fourier equation in monodimensional cylindrical geometry is applied to the three fuel zones: pellet, gap and cladding. The main assumption of the model is to consider only the radial heat transfer, thus disregarding both the axial and the circumferential diffusions.

For the pellet, gap and cladding we write:

$$\rho_p\, c_{p,p} \frac{\partial T_p}{\partial t} = \frac{1}{r} \frac{\partial}{\partial r} \left(r\, k_p \frac{\partial T_p}{\partial r} \right) + q''' \qquad (4)$$

$$\frac{\partial}{\partial r} \left(r\, k_g \frac{\partial T_g}{\partial r} \right) = 0 \qquad (5)$$

$$\rho_c\, c_{p,c} \frac{\partial T_c}{\partial t} = \frac{1}{r} \frac{\partial}{\partial r} \left(r\, k_c \frac{\partial T_c}{\partial r} \right) \qquad (6)$$

These equations, together with the conditions of heat flux vanishing at the pellet centre and the continuities of the temperatures

and heat fluxes at the three boundaries pellet-gap-cladding-coolant allow determination of $T_p(r,t)$, $T_c(r,t)$ and $T_g(r,t)$.

In particular, the condition at the cladding-coolant interface reads

$$-k\left.\frac{\partial T_c}{\partial r}\right|_{r=Roc} = H\left(T_{Roc} - \overline{T}_m\right) \tag{7}$$

in which the rhs of the above equation refers to the mean value $\overline{T}_m$ of the water temperature over the entire core height *L*, i.e.:

$$\overline{T}_m = \frac{\int_0^L T_m(z)\,dz}{L} \tag{8}$$

The coolant temperature profile $T_m(z)$ is obtained from the solution of the thermalhydraulics model below detailed.

In addition to the above equations, we also used 5 correlations synthesising the dependences of $c_{p,p}, c_{p,c}, k_p, k_c$ as a function of the temperature and k_g as a function of both the reactor power and the burn-up.

3.3 Thermalhydraulics

The core is simulated by a single channel representing the average of the three channels mentioned in the Neutron Dynamics subsection. The coolant is assumed to consist of an incompressible but thermal expandable water; in case of two-phase flow, thermal equilibrium is also assumed. In terms of flow rate, enthalpy and pressure through the heated channel, the balance equations for mass, energy and momentum conservation read

$$\Omega\frac{\partial \gamma_m}{\partial t} + \frac{\partial \Gamma_m}{\partial z} = 0 \tag{9}$$

$$\rho_m\frac{\partial h_m}{\partial t} + \frac{\Gamma_m}{\Omega}\frac{\partial h_m}{\partial z} = \frac{q''\Pi_h}{\Omega} \tag{10}$$

$$\frac{1}{\Omega}\frac{\partial \Gamma_m}{\partial t} + \frac{1}{\Omega^2}\frac{\partial}{\partial z}\left(\frac{\Gamma_m^2}{\rho_m}\right) = -\frac{\partial p}{\partial z} - \gamma_m g\cos\vartheta - f\frac{2\Gamma_m\left|\Gamma_m\right|}{D_e\Omega^2\gamma_m} \tag{11}$$

Notice that the energy balance equation is written by neglecting the axial conduction, the viscous dissipation term and the pressure variation. The system of the 3 above equations is completed by a set of heat transfer, pressure drop and water properties correlations. The latter allow us to transform the enthalpy distribution $h_m(z)$ along the core height into the corresponding temperature distribution $T_m(z)$, used to obtain the mean value $\overline{T}_m$ via eq.(8). The mean density value $\overline{\gamma}_m$ is also computed from a similar equation applied to $\gamma_m(z)$. In summary, the thermalhydraulics is modeled by a set of 6 equations in the 6 unknowns.

3.4 Numerical solution of the complete model equations

The time profiles of the process variables required for the ANN training have been obtained by numerically solving with a fixed time step Δt=0.1s the system of 13 partial differential equations and the above mentioned 8 correlations describing the core transients. This solution will be here called the solution of the complete model.

At the generic time *t* in addition to the assigned forcing functions $T_{m,i}(t)$, $\Gamma_m(t)$ and $\rho_{CR}(t)$ which drive the transients, we know all the process variables, which have been computed in the preceding time step. The model equations integration is carried out along the following steps:

1. First of all we solve the equations reported in the Thermalhydraulics subsection thus obtaining $\overline{T}_m$ and $\overline{\gamma}_m$ at time $t+\Delta t$; then the corresponding reactivity contributes $\rho_{\overline{T}m}$ and $\rho_{\overline{\gamma}m}$ appearing in eq.(1) are computed. This system was solved with a fixed time step of 0.1s. The total reactivity $\rho(t+\Delta t)$ is then obtained by adding the Doppler term in correspondence of $T_f(t)$.

2. At this point we may solve the equations of the neutron dynamics thus obtaining the neutron and precursors concentrations at $t+\Delta t$ and then the reactor power $W_n(t+\Delta t)$. This solution was obtained with a variable time step around 10ms;

3. Finally, we solve the fuel heat transfer equations and obtain $T_f(t+\Delta t)$ and the thermal power (i.e. the power transferred from fuel to coolant) $W_{th}(t+\Delta t)$: to do this, the fuel is subdivided into five radial regions, three for the pellet and the remaining two for the gap and the cladding; by assuming a parabolic temperature profile within the pellet we obtain a system of algebraic equations, easily solvable by matrix inversion.

Knowledge of all the above quantities allows us to proceed one step further in time.

3.5 Simplified balance

As we shall see below, the core transient computations may be carried out much faster and with an acceptable loss of accuracy if some of the ANN input variables are obtained by solving a simplified thermalhydraulic model. Therefore we have set up another model in terms of effective variables $\overline{\Theta}_m, \Theta_{f,eff}$ and Q_{th} which take the place and the physical meaning of $\overline{T}_m$, $T_{f,eff}$ and W_{th}, respectively. In the simplified model the equations describing the energy balances in the fuel and in the coolant and the heat exchange between them, are

$$M_m C_{m,eff}\frac{d\overline{\Theta}_m}{dt} + 2\Gamma_m C_{m,eff}\left(\overline{\Theta}_m - \Theta_{m,i}\right) = Q_{th} \tag{12}$$

$$M_f C_{f,eff}\frac{d\Theta_{f,eff}}{dt} = W_n - Q_{th} \tag{13}$$

$$Q_{th} = U_{eff}\left(\Theta_{f,eff} - \overline{\Theta}_m\right) \tag{14}$$

The link with the complete model is established by forcing $\overline{\Theta}_m, \Theta_{f,eff}$ and Q_{th} to be close to $\overline{T}_m$, $T_{f,eff}$ and W_{th} respectively. This is obtained by suitably discretizing the above differential

TABLE I
ANN #1 for neutron power and precursor concentrations prediction - Inputs and Outputs description

Inputs	
1. $\Theta_{m,i}^{t+\Delta t} - \Theta_{m,i}^{t}$	9. $\Gamma_m^{t+\Delta t} / Q_{th}^{t} - \Gamma_m^{t} / Q_{th}^{t-\Delta t}$
2. $\Gamma_m^{t+\Delta t} / \Gamma_m^{t} - 1$	10. $\left(Q_{th}^{t} - W_n^{t}\right) - \left(Q_{th}^{t-\Delta t} - W_n^{t-\Delta t}\right)$
3. $\Gamma_m^{t+\Delta t} / Q_{th}^{t}$	11. $W_n^{t} - W_n^{t-\Delta t}$
4. $Q_{th}^{t} - W_n^{t}$	12. $\overline{\Theta}_m^{t} - \overline{\Theta}_m^{t-\Delta t}$
5. $W_n^{t} / W_n^{t-\Delta t} - 1$	13. $\Theta_{f,eff}^{t} - \Theta_{f,eff}^{t-\Delta t}$
6. W_n^{t}	14. $\rho_{CR}^{t+\Delta t}$
7. $\overline{\Theta}_{m,i}^{t}$	15., 20. c_j^{t} / n^{t} (with $j = 1, 6$)
8. $\Theta_{f,eff}^{t}$	
Outputs	
1. $W_n^{t+\Delta t} / W_n^{t} - 1$	
2., 7. $\left(c_j / n\right)^{t+\Delta t} / \left(c_j / n\right)^{t} - 1$ (with $j = 1, 6$)	

TABLE II
ANN #2 for U_{eff} prediction - Inputs and Outputs description

Inputs	
1. $\Theta_{m,i}^{t+\Delta t} - \Theta_{m,i}^{t}$	7. $W_n^{t} - W_n^{t-\Delta t}$
2. $\Gamma_m^{t+\Delta t} / \Gamma_m^{t} - 1$	8. $\overline{\Theta}_m^{t} - \overline{\Theta}_m^{t-\Delta t}$
3. $Q_{th}^{t} - W_n^{t}$	9. $T_{f,eff}^{t} - T_{f,eff}^{t-\Delta t}$
4. W_n^{t}	10. $\Gamma_m^{t+\Delta t}$
5. $\overline{\Theta}_m^{t}$	11. U_{eff}^{t}
6. $\Theta_{f,eff}^{t}$	
Outputs	
1. $W_n^{t+\Delta t} / \left(\Theta_{f,eff}^{t+\Delta t} - \overline{\Theta}_m^{t+\Delta t}\right) = U_{eff}^{t+\Delta t}$	

equations and then by considering $C_{m,eff}$, $C_{f,eff}$ and U_{eff} as effective time dependent functions, to be determined by the condition that the numerical values of $\overline{T}_m$, $T_{f,eff}$ and W_{th} computed with the complete model should satisfy the simplified model too, suitably discretized. In correspondence of the process variables computed with the complete model driven by the forcing functions of Fig.2, it turns out that:

i) $C_{f,eff}$ is almost constant;

ii) $C_{m,eff}$ essentially only depends on $\overline{T}_m$ and it may be fitted by the simple linear expression

$$C_{m,eff} = 5.635 + \left(\overline{T}_m - 300\right) \cdot 0.0214;$$

iii) U_{eff} does not lend itself to a simple analytical expression, probably because it depends on too many variables. Therefore we preferred to compute it via a dedicated ANN as detailed below.

4. THE NETWORK

As usual in the numerical approaches, the ANN solution to a system of equations is found by means of a succession of computations in correspondence of a sequence of discrete times. At each time the input nodes are fed with the values computed at the preceding step and the output nodes release the corresponding values one step ahead, which represent the ANN input values for the next calculation. However, unlike the classic numerical approaches, the ANN methodology suffers of what is called the "problem representation" namely the selection of the input variables, which is a matter least guided by useful theory. Indeed, in our experience, the ANN approach almost always fails if it is pursued with as many input and output nodes as the number of dependent variables which appear in the model equations and also if just the values of these variables are worked out by the network. Instead, we think that the following two rules should guide the design of an ANN solution:

Rule 1: the input quantities should be suitable functions of the dependent variable values, such as differences or ratios of present values and preceding ones, or between pairs of variables. In other words, a suitable pre-processing of the variables is appropriate.

Rule 2: the output quantities should be only a part of the dependent variables themselves (one step ahead) or of some functions of them which allow simple analytical computation of the remaining ones.

We are aware of the fact that these are very vague rules but we are also aware that at present time, the ANN computing is more an art than a well established technique.

In the problem at hand, after many trials, we selected an ANN with 20 input, 16 hidden and 7 output nodes. The input and output variables at time t are reported in Table I. The network was trained by 4000 patterns obtained with the system driven by the randomly generated forcing functions appearing in Fig.2. Notice that the input values $\Theta_{m,i}$, Γ_m and ρ_{CR} are evaluated at time $t + \Delta t$: this is correct since they are the known forcing functions which drive the system. The output quantities are essentially the reactor power W_n and the precursor concentrations at $t + \Delta t$. Thus the ANN essentially gives the solution to the neutron dynamics.

In agreement with the Rule 2, the remaining values of the process variables are then analytically computed with reference to the simplified model by solving the system of 3 algebraic equations resulting from the discretization of the equations in Section 3.5.

Concerning the value of U_{eff} appearing in the last of these equations we already remarked that we have not been able to find a simple analytical expression which gives U_{eff} in terms of the other variables. Therefore it appeared convenient to extract this relationship by resorting to another ANN. After some trials, we choose a network with 14 input, 4 hidden and 1 output nodes,

whose inputs and outputs are reported in Table II. The training of this new ANN turned out to be particularly difficult so that, in addition to the 4000 patterns used for the first ANN, we added 14000 new patterns extracted from the process variables time profiles computed with the complete model in correspondence of the system driven by the forcing functions reported in Fig.3.

5. NUMERICAL RESULTS

The reliability and accuracy of the ANN results have been checked by considering the following four transients computed with the TRAP code mentioned in Section 3.

5.1 Load following transient

A step descent to 75% of the nominal grid load requested to the plant is imposed at t=0. The control logic of the plant reacts to comply with demand by acting on the turbine admission valve and on the control rods: together with the reactor feedback effects, the reactor power reaches the appropriate and steady value within few hundred seconds (Fig.4).

5.2 Bypass line valve failure

At t=0 a bypass line valve toward the condenser fails open. The steam generator undergoes a sudden depressurisation and owing to the subsequent large descent of the core inlet temperature, the reactor power increases. Even the scram system is supposed to fail. After 120 s, the bypass valve is partially closed and the reactor moves to a steady value, with a power 10% higher than the nominal one (Fig.5).

5.3 Loss of flow accident

A main coolant pumps trip (two pumps out of four) occurs: the core flow rate decreases to 70% of the nominal value, while the reactor power follows a damped oscillation until a steady condition is reached, 7% less than the initial power (Fig.6).

5.4 Inadvertent control rod extraction accident

The transient starts with a rampwise partial extraction of a control rod, which inserts a positive reactivity of about 300 pcm in 2.5 s. At a delay time of 10 s the trip of two pumps out of four occurs. The resulting loss of flow determines a further increase of the fluid temperature which counteracts the reactor power increase due to the rod extraction. Finally a steady condition 8% below the nominal power (Fig.7) is reached.

All the above tests have been done by using home-made FORTRAN codes either for the integration of the complete model and for the two ANNs plus the analytical solution of the three algebraic equations of the simplified model. It turned out that, on the average, the ANN solution is about 20 times faster than the numerical one.

6. CONCLUSIONS

The dynamic PSAs, in which the time evolution of the process variables is taken into account, is conceptually preferable to the classic, static PSAs. However every benefit has a cost and here the cost is the requirement of a much larger CPU time. This is particularly important in the Monte Carlo approach, where a very large number of stories have to be processed before significant statistical results are obtained. In connection with the Monte Carlo approach the ANN displays its advantages since the time required for performing the deterministic calculations involved in the dynamic approach is very low in comparison with that of the numerical integration of the relevant equations. This advantage is paid with the loss of precision, typical of the ANN approach.

In the present paper we considered the possibility of an ANN computation of the core dynamics. To this aim we considered a code which simulates a PWR plant and substituted the routines which numerically integrate the core equations with an ANN routine. Numerical results relating to four plant transients indicate a computer time reduction of about 20. In the examined transients the ANN approach seems to be capable of obtaining a reasonable approximation, which may be regarded as acceptable when compared to the degree of uncertainty typical of the reliability data.

NOMENCLATURE

Symbols:

A	neutronic feedback coefficient
C	coefficient
c_j	j-th neutron precursor concentration (j=1, 6)
c_p	specific heat at constant pressure
D	diameter
f	friction coefficient
g	gravitational constant
h	enthalpy
H	heat transfer coefficient
k	thermal conductivity
L	core length
M	mass
p	pressure
q''	surface heat flux
q'''	power density per unit volume of fuel
r, R	radius
t	time
T, Θ	temperature
U	heat transfer parameter
W, Q	power
z	abscissa

Greeks:

ϑ	inclination
γ	density
ρ	reactivity
Ω	cross section area
Π	cross section perimeter
Γ	mass flow rate

Subscripts:

c	cladding
CR	control rod
d	density
e	equivalent
eff	effective
f	fuel
g	gap
h	heated
i	inlet
m	moderator
oc	outer cladding
p	pellet

REFERENCES

Cheon Se-Woo, Chang Soon-Heung and Chung Hak-Yeong, Nam Bien Zeung, 1993, "Development Strategies of an Expert System for Multiple Alarm Processing and Diagnosis in Nuclear Power Plants", *IEEE Trans. on Nucl. Sci*., Vol 40, No 4.

Cichocki, A. and Unbehauen, R., 1993, Neural Networks for Optimization and Signal Processing, J.Wiley & Sons Inc, NY.

Ciftcioglu, O., Turkcan, E. and Seker, S., 1991, "Failure Detection Studies by Layered Neural Network*," ASME Conference on Neural Networks*, San Diego, USA.

Devooght, J. and Smidts, C., 1992a, "Probabilistic reactor dynamics. I. The theory of continuous event trees*", Nuclear Science and Engineering*, 111, pp. 229-240.

Devooght, J. and Smidts, C., 1992b, "Probabilistic reactor dynamics. II. A Monte Carlo study of a fast reactor transient"*, Nuclear Science and Engineering*, 111, pp. 241-256.

Gelbard, E. M. and Spanier, J., 1969, Monte Carlo Principles and Neutron Transport Problems, Addison Wesley, Reading, MA.

Guo, Z. and Uhrig, R.E., 1992, "Nuclear Power Plant Performance Study by Using Neural Networks"*, IEEE Trans. on Nucl. Science*, Vol 39, No 4, pp. 915-918.

Jouse, W. and Williams, J., 1992, "Neural Network Control of Anomalous Reactor Transients", *Trans. Am. Nucl. Soc*., 66, 115.

Jouse, W. and Williams, J., 1993, "Safety Control of Nuclear Power Operations Using Self-Programming Neural Networks", *Nuclear Science and Engineering*., 114, 42-54.

Kalos, M. H. and Whitlock, P. A., 1986, Monte Carlo Methods Vol. 1: Basics, John Wiley, New York.

Kamat, S. J. and Franzmeier, W. E., 1976, "Determination of reliability using event-based Monte Carlo simulation Part II," *IEEE Trans. on Reliability*, R-25 (1), pp. 254-255.

Kamat, S. J. and Riley, M. W., 1975, "Determination of reliability using event-based Monte Carlo simulation*," IEEE Trans. on Reliability*, R-24 (1), pp. 73-75.

Kim Han Gon, Chang Soon-Heung and Lee Byung Ho, 1993, "Pressurized Water Reactor Core Parameter Prediction Using Artificial Neural Network", *Nuclear Science and Engineering,* 113, 70-76.

Kumamoto, H., Tanaka, K. and Inoue, K., 1977, "Efficient evaluation of system reliability by Monte Carlo Method", *IEEE Trans. on Reliability*, R-26 (5), pp. 311-315.

Lewis, E. E. and Bohm, F., 1984, "Monte Carlo simulation of Markov unreliability models", *Nuclear Engineering and Design*, 77, pp. 49-62.

Lewis, E. E. and Zhuguo, T., 1986, "Monte Carlo reliability modeling by inhomogeneous Markov processes"*, Reliability Engineering & System Safety*, 16, pp. 277-296.

Marseguerra, M. and Zio, E., 1992, "Approaching Dynamic Reliability by Monte Carlo Simulation", NATO ASI Series F, vol. 120, pp. 44-58.

Marseguerra, M. and Zio, E., 1993, "Dynamic PSA by Monte Carlo Event Simulation Analysis*", Proceedings of the PSA'93 International Meeting*, pp. 615-624, Jan. 1993.

Marseguerra, M. and Zio, E., 1994a, "Fault Diagnosis Via Neural Networks: the Boltzmann Machine", *Nuclear Science and Engineering*, 117, No 3, 194-200.

Marseguerra M. and Zio E., 1994b, *Ann. Nucl. Energy*, Vol 21, No 4, 249-258.

Muller, B. and Reinhardt, J., 1991, Neural Networks: an Introduction, Springer Verlag, Berlin.

Parks, D. C., El-Sharkawi, M. A., Marks II, R. J., Atlas, L. E. and Damborg, M. J., 1991, "Electric load forecasting Using an Artificial Neural Network," *IEEE Trans. Power Syst*., Vol 6, No 2, 442-449.

Ricotti, M.E., 1994, "A Model for Operability and Safety Analysis of Conventional and New Conception PWRs", Ph.D. Thesis, Polytechnic of Milan, Milan.

Roh Myung-Sub, Cheon Se-Woo and Chang Soon-Heung, 1991, "Thermal Power Prediction of Nuclear Power Plant Using Neural Network and Parity Space Model*," IEEE Trans. on Nucl. Sci*., Vol 38, No 2.

Rumelhart, D. E. and McClelland, J. L., 1986, Parallel Distributed Processing, Vol.1, MIT Press, Cambridge, MA.

Sanchez-Sinencio, E. and Lau, C., 1992, Artificial Neural Networks, IEEE Press, NY.

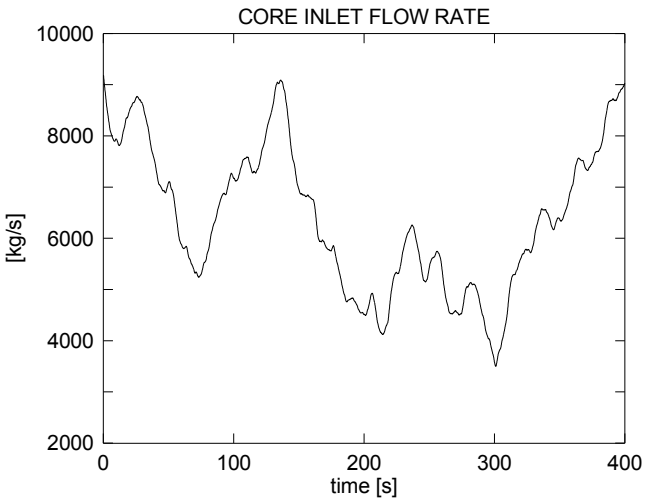
Siu, N. O., 1994, "Risk Assessment for Dynamic Systems: An Overview", *Reliability Engineering & System Safety*, 43, 1, pp. 43-73.

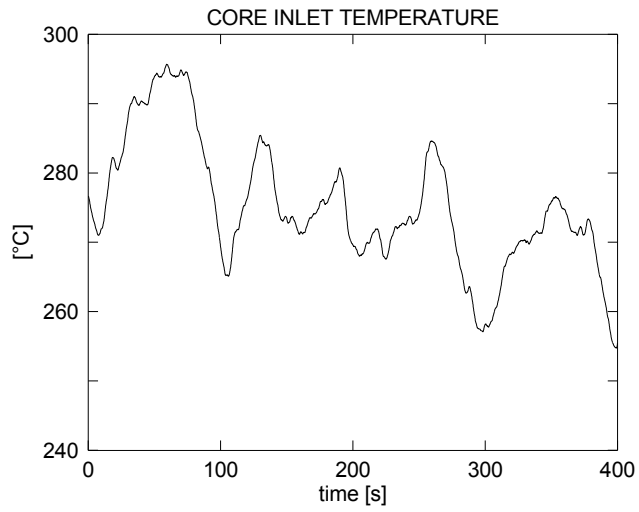
Siu, N. O. and Deoss, D. L., 1989, "A simulation model for dynamic system availability analysis", MITNE-287, Massachusetts Institute of Technology, Cambridge, MA.

U.S. Nuclear Regulatory Commission, 1981, PRA Procedures Guides, NUREG/CR-2300.

Uhrig, R. E., 1991, "Potential Application of Neural Networks to the Operation of Nuclear Power Plants," *Nuclear Safety*, Vol 32, No 1, pp. 68-79.

White, H., 1989, "Learning in Neural Networks: a Statistical Perspective", *Neural Comp*., i, 425-464.





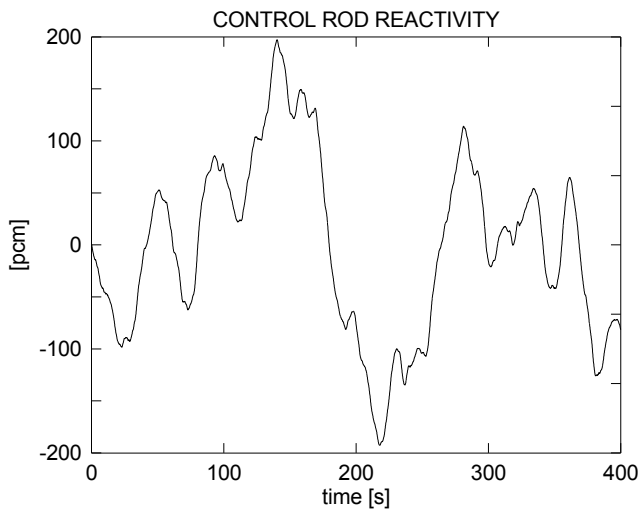
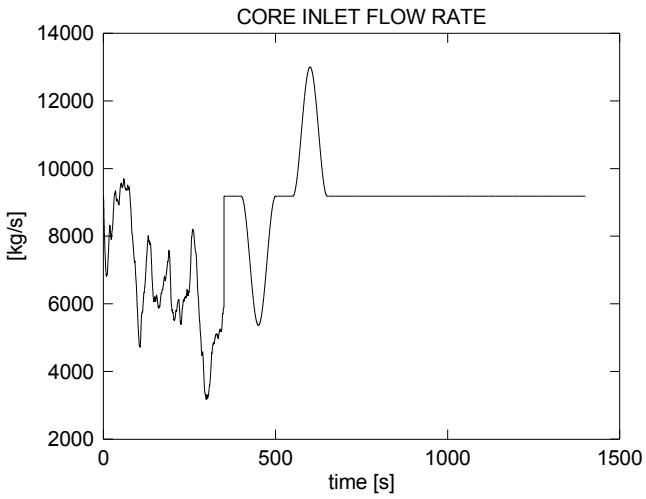
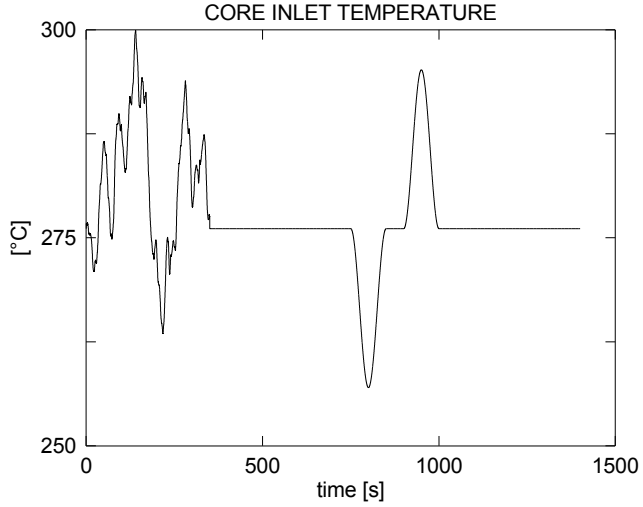


Fig. 2. Forcing functions used for ANN #1 training.





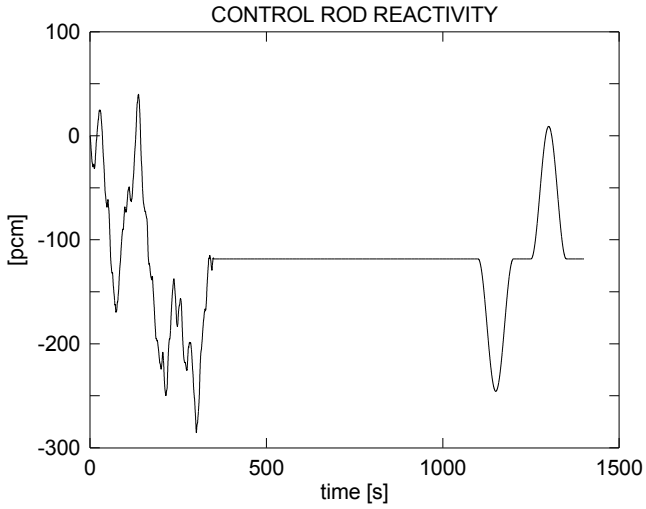


Fig. 3. Forcing functions used for ANN #2 training.

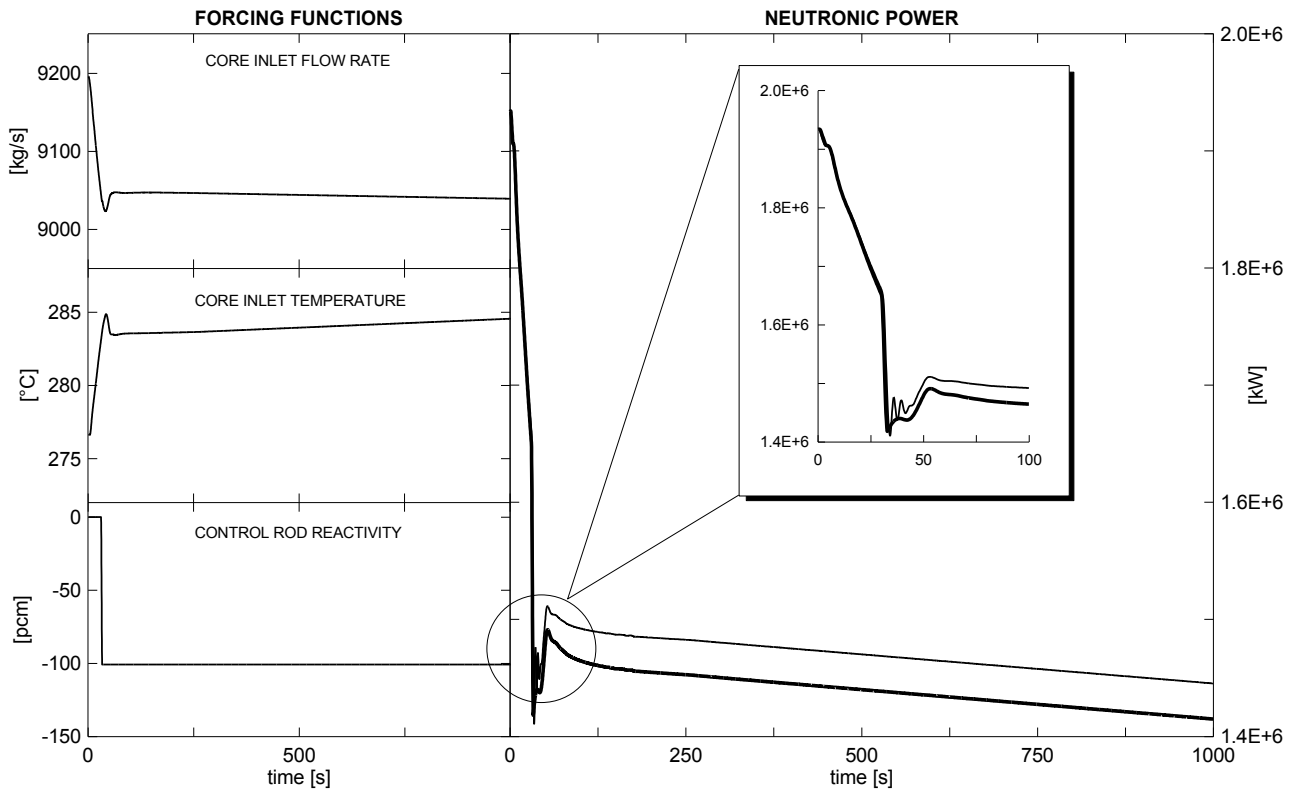


Fig. 4. Load following test results. In the right side: ANN prediction (thin line) vs. TRAP routines computation (thick line).

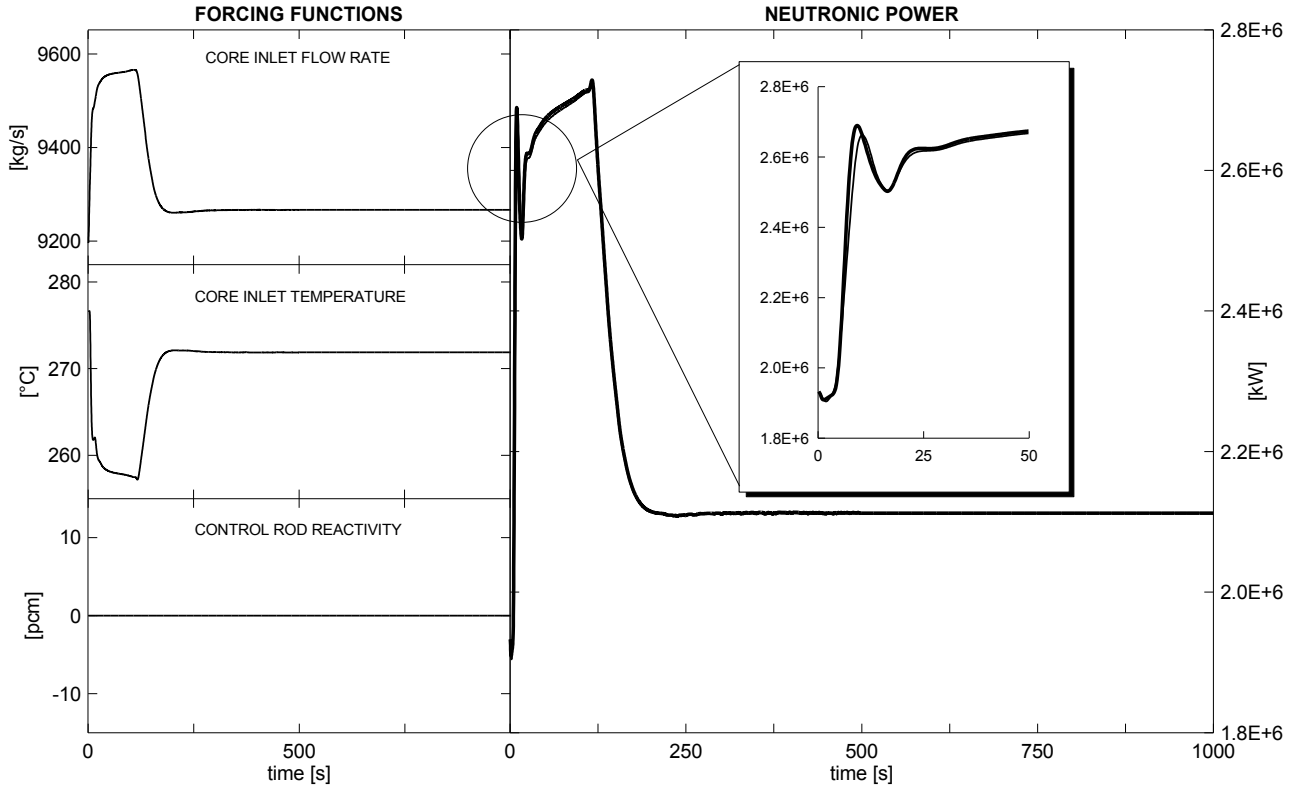


Fig. 5. Bypass line valve failure test results. In the right side: ANN prediction (thin line) vs. TRAP routines computation (thick line).

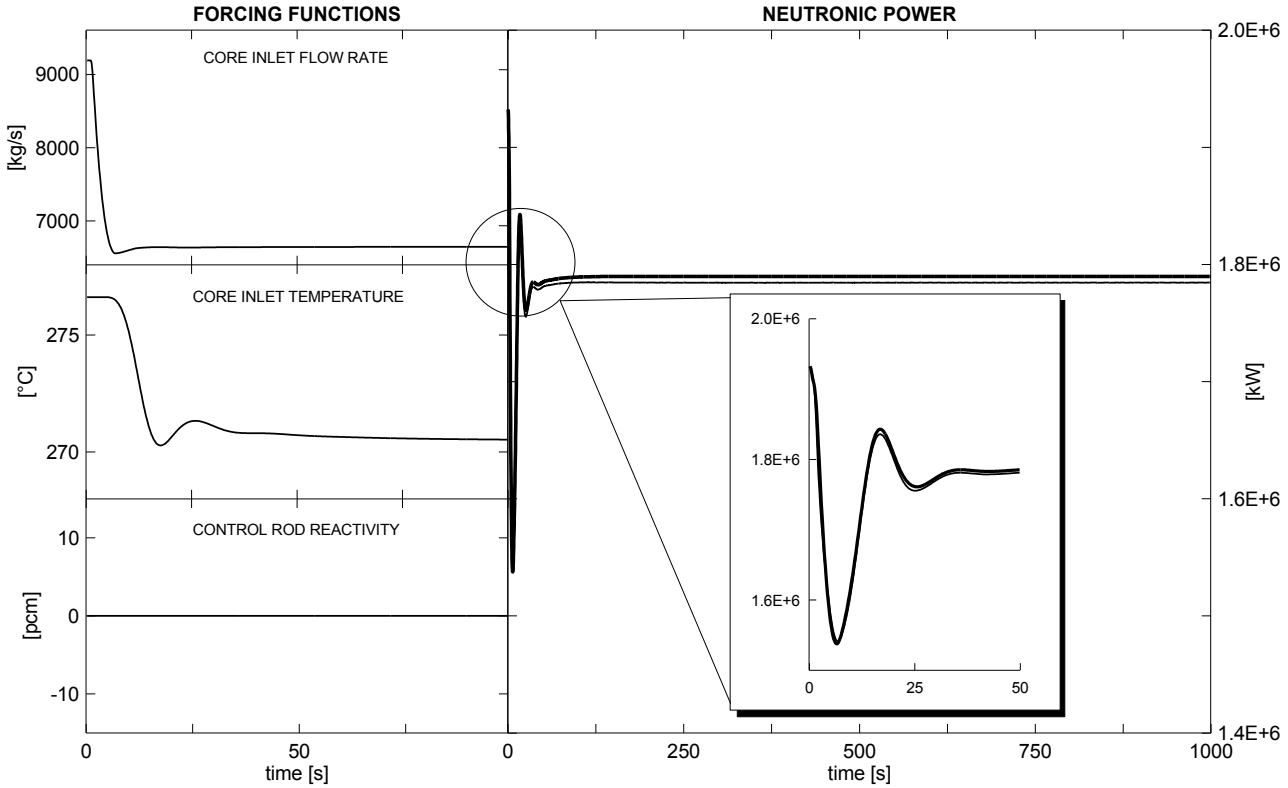


Fig. 6. Loss of flow test results. In the right side: ANN prediction (thin line) vs. TRAP routines computation (thick line).

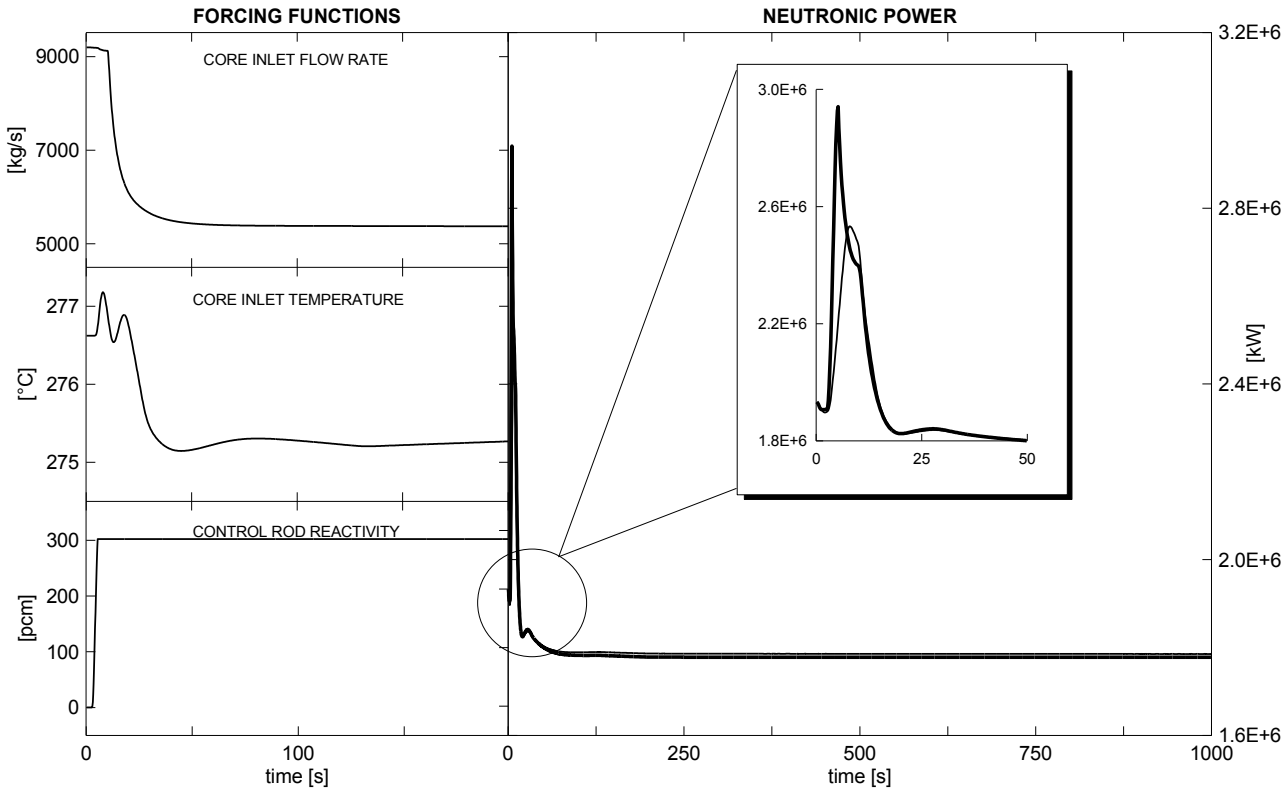


Fig. 7. Inadvertent control rod extraction test results. In the right side: ANN prediction (thin line) vs. TRAP routines computation (thick line).