This is a post-peer-review, pre-copyedit version of an article published in 19th Polish Control Conference. The final authenticated version is available online at: http://dx.doi.org/10.1007/978-3-319-60699-6_33

Postprint of: Juszczuk D., Tarnawski J., Karla T., Duzinkiewicz K., Real-Time Basic Principles Nuclear Reactor Simulator Based on Client-Server Network Architecture with WebBrowser as User Interface. In: Mitkowski W., Kacprzyk J., Oprzędkiewicz K., Skruch P. (eds), Trends in Advanced Intelligent Control, Optimization and Automation. KKA 2017. Advances in Intelligent Systems and Computing, vol 577 (2017), pp. 344-353, Springer, https://doi.org/10.1007/978-3-319-60699-6_33

Real-Time Basic Principles Nuclear Reactor Simulator Based on Client-Server Network Architecture with WebBrowser as User Interface

Dymitr Juszczuk, Jaroslaw Tarnawski, Tomasz Karla, Kazimierz Duzinkiewicz

Department of Control Engineering, Faculty of Electrical and Control Engineering, Gdansk University of Technology, Gdansk, Poland
dymitrjuszczuk@gmail.com, jaroslaw.tarnawski@pg.gda.pl, tomasz.karla@pg.gda.pl, kazimierz.duzinkiewicz@pg.gda.pl

Abstract. The real-time simulator of nuclear reactor basic processes (neutron kinetics, heat generation and its exchange, poisoning and burning up fuel) build in a network environment is presented in this paper. The client-server architecture was introduced, where the server is a powerful computing unit and the web browser application is a client for user interface purposes. The challenge was to develop an application running under the regime of real-time, with a high temporal resolution, in an environment which is not a native real-time. The problem of a real-time operation taking into account the variable time of calculations and a communication latency was solved using the developed mechanism of step length adaptation. Results of multiple studies of a numerical compliance with the reference simulator proved correctness of the developed application.

1 Introduction

Simulators can be used for educational and dissemination purposes supporting the classical teaching process. Real-time simulators have the additional property because they offer a familiarization with the dynamics of processes and their temporal interrelationships. The basic principles nuclear reactor (NR) simulator presented in this paper is designed to work in a real-time based on a network environment and a web browser which does not work natively in a real-time regime. The real-time simulator is expected to deliver simulation results at certain moments in time, on-line, with a reference 1:1 to a real-time. A simulation time resolution associated with a simulator step is a very important parameter. Acquisition of input signals, mathematical calculations and presentation of simulation variables in one simulator iteration shall be performed faster than a simulation step.The real-time simulator needs a fast, time-efficient solver able to generate results in a finite, predictable time to produce the results with a prescribed temporal resolution. The deterioration in the accuracy of calculations is expected comparing to non real-time simulators, but a numerical stability must be ensured.

The subject of non real-time simulation of nuclear reactor basic principles can be found among others in [1]. The conception of the real-time simulation with a web browser as a user interface can be found e. g. in [2] but this application is not used with processes requiring a high computing performance like NR processes.

Authors previously developed a cross-platform real-time NR simulator targeted for different hardware-software platforms (like PC or RaspberryPi) [3]. In the following paper [4] authors introduced the algorithm for maintaining a real-time operation in case of delays in simulations in non real-time environments. The usage of this method on different hardware platforms revealed the restrictions on the length of a simulation step and the need for estimating a maximal length of a simulation step for specific hardware-software platforms [5]. Previous works were concerned only about a standalone simulator, without the usage of a network, a web browser or simultinous access of many users. The comprehensive theory on a real-time simulation with many applications in different areas can be found in the book [6].

2 Physical and control processes in the nuclear reactor and their temporal characteristics

One of the simplest models but well reflecting the nature of the processes in the NR is the point model [7], in which all variables are averaged over the volume of the core. Figure 1 presents processes and control effects of NR included in presented simulators. All of these processes have different time scales. Table 1 summarizes the various processes and their transition times.

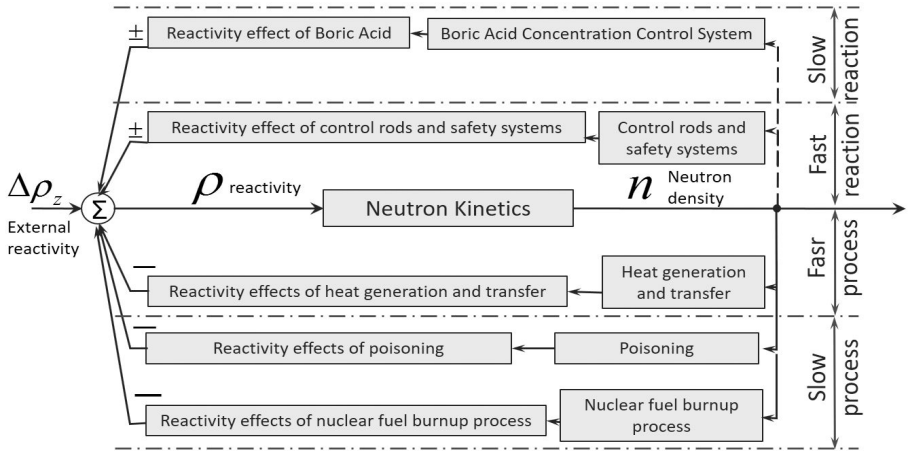

Fig. 1. Main processes occurring in a nuclear reactor affecting its state

Another approach to the modeling of nuclear processes and equipment can be found e.g. in [8], [9]. The comprehensive information about reactor physics and related topics can be found in [10], [11].

Table 1. Processes occurring in the reactor and their duration of transient states

The process	The duration of transient states
Neutron kinetics	1E-5 to 1E-3 seconds
Heat transfers between fuel and coolant/ moderator, Changes in reactivity resulting from changes in temperatures of the fuel and the coolant/ moderator	3 to 6 minutes
Changes in reactivity caused by changing positions of control rods	up to 125 s
Changes in reactivity caused by changing the concentration of boric acid in the moderator	several hours
The xenon poisoning occurring while changing the power level of the reactor	up to 60 hours
The samarium poisoning	up to 60 hours
The nuclear fuel burnup	tens of days and months

3 Simulator design assumptions and requirements

The aim of the study was to develop a real-time basic principle nuclear reactor simulator operating in the regime of real-time in a network environment that can support the interface prepared in a web browser. Specific assumptions include:

- an implementation of mathematic point model of nuclear reactor basic processes,
- a real-time operation in a network environment with the usage of Internet network,
- the ability to present the results to the users and input control signals to the simulator from multiple clients simultaneously,
- the user interface in the form of the web browser application,
- the ability to change while in a simulation following parameters: the level of the control rods, the density of boric acid, the coolant temperature at the inlet to the reactor,
- the numeric compatibility with the reference simulator build in the MATLAB/Simulink environment with the ability of archive data in a form compatible with the MATLAB for further processing,
- the software-hardware independence on the client side,
- the client-server topology with the use of the widely available Ethernet and the TCP protocol.

Fulfilling all the assumptions, especially working in the regime of real-time in a networked environment required development of a mechanism to adapt the computation step that would provide a real-time simulation in environments without real-time kernel even in a case of a large load on the server and the network at the expense of compromising the accuracy of the calculations.

4 The issue of nuclear reactor mathematical model real-time calculations in a network environment

The mathematical model mentioned in the previous chapter consists of 18 differential equations and several algebraic equations. Due to different scales of dynamics the problem called stiffness can be expected during calculations. In [12] it can be read "An ordinary differential equation problem is stiff if the solution being sought is varying slowly, but there are nearby solutions that vary rapidly, so the numerical method must take small steps to obtain satisfactory results". Unfortunately, exactly the kind of situation was faced during the construction of the basic principle nuclear reactor simulator. To calculate the results of the differential equations, the stiff or non-stiff methods can be used. Again quoting [12] "Stiffness is an efficiency issue. If we were not concerned with how much time a computation takes, we would not be concerned about stiffness. Non-stiff methods can solve stiff problems; they just take a long time to do it". Unfortunately, in real-time simulations implementation of both stiff or non-stiff methods with very small steps and long calculations cannot be applied. Methods with a fixed or variable step can be used to solve systems of differential equations. Constant step methods are easier to implement. In this article the method with a variable step was applied not only from the need to step reduce but also to take account of major changes and to avoid stiffness. In a non real-time environment unexpected incidental delays may occur due to e.g. communication, data loading and operating system interrupts. Then there may be situations where the computation time for a single-step of a simulation exceeds a prescribed simulation step. To maintain the synchronization with a simulation time lengthening simulation steps is used - in order to catch up a simulation time. The multi-step catch up simulation time algorithm is described in the following section. The most commonly used methods for solving differential equations are the Euler or Runge-Kutta methods. Higher order methods are numerically stable and more accurate. Taking into account all the above issues the variable step Runge-Kutta (specifically forth order) method was applied in the simulator.

5 The mechanism of a simulation step adaptation in the web simulator

The real-time supervision mechanism was located on a client side to ensure that the user gets simulation results on time. The time between sending the request of the results from the server and getting answer is counted by the supervision

mechanism. This value is used to determine if current data is on time. The algorithm of step adaptation will be used when the mechanism detects the delay in data delivery. The delay means difference between the simulation time and the real-time. It must be compensateted by algorithm. The basis for the development of a step adaptation mechanism is presented in [4]. The idea assumes that in case of a simulation delay, the next simulation step will be as long as the previous cycle time. It is assumed that occurred delay will cause the same communication time extension in the next program cycle. The new simulation step is then extended to multiplicity of the base step to maintain the real-time regime. The principle of step adaptation is shown on fig.2 chart a).

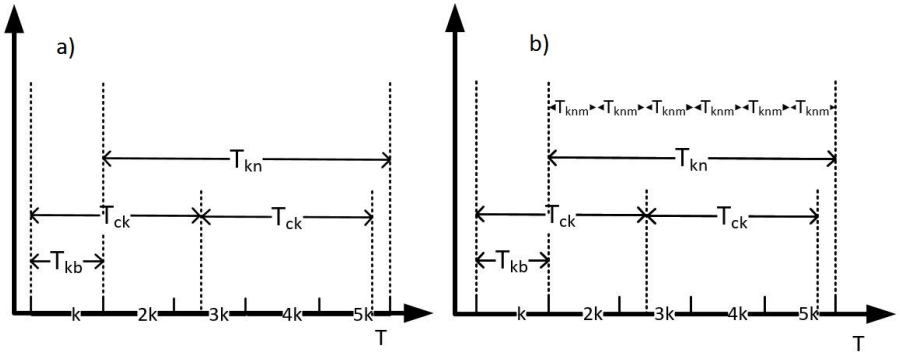

Fig. 2. Charts showing the idea of step adaptation

The base steps are indicated on the x-axis. The communication starts at the beginning of the k period. The cycle of the program should be shorter than the established simulation step T_{kb}. The cycle, however, exceeds simulation step and is T_{ck}, almost two and half-step base. Assuming that the next cycle time also will be T_{ck}, the next calculation point was set at the end of $5k$ period. The new simulation step is the period from the end of the last step to a new calculation point (T_{kn}). A new simulation step has the length of four base steps. Such a step would compensate the delay between the simulation time and the real-time. However, the studies have shown that eliminating the delay in one step has a big impact on the quality of the simulation results and can lead to a numerical instability. To prevent that, the new simulation step is calculated by an operation that divides the base simulation step T_{kn} on the m shorter periods (shown on fig.2 chart b)). This operation will achieve the final step in several cycles of a simulation and reduce the negative effects of its changes. The current step will be increased by T_{knm} period in each simulation cycle up to the final step T_{kn}.

6 The realization of network simulator

6.1 Hardware-Software platform of simulator

After considering all the requirements and objectives software-hardware environment for the implementation of the simulator was selected. The hardware platform for the simulator server was a mainstream Intel i7 PC class computer with the Ubuntu 14.0 operating system. Such configuration has been chosen because of their popularity, availability, reliability and software security in server environments. In addition, the Node.js environment was used which made it possible to ensure the real-time communication with clients that use web browsers. The only requirements on the client side were access to the Internet network and a modern web browser.

The client hardware-software platform can be any device that utilize Ethernet networks (including mobile, e.g. using Wi-Fi) which are equipped with a web browser compliant with modern standards of displaying Web pages (the JavaScript is required).

6.2 Web communication methods used in simulator

It is assumed that only one client device is a real-time maintaining unit in a simulation. The user interface has authorisation system with three types of users. Simulation can be configured and run only by the user with full permissions (administrator). The rest of the users participate in the real-time simulation maintained by the administrator unit. However, non administrator users have the ability to change the control values and have an impact on the simulation.

In order to achieve the two-way real-time communication between clients and the server, WebSocket (WS) technology was used. WS technology offers standard virtual channels, allowing the server to communicate with some or all users. The protocol creates a "Socket" in a browser that have the IP address and the port, maintains a two-way, simultaneous communication between the client and the server. The technology uses the TCP transport layer and allows to bypass the interpretation of messages over HTTP. WS reduces the unnecessary network traffic and shortens the waiting time for the information in the relation to conventional methods of a communication, because if the communication is once established the server can send informations at any time, as soon as they are ready to transfer.

6.3 The user interface of web simulator

The user interface was made as a web browser application. Static elements on the site were designed using HTML5. Animations, an event handling and data operations were developed in the JavaScript. The user interface was designed to present as much relevant information and be able to adjust the displayed objects to the client needs. User interface is presented on fig.3.

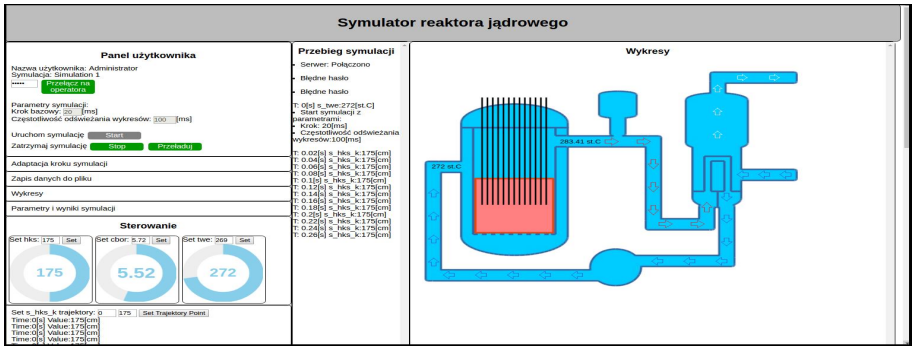

Fig. 3. The user interface

The interface consists of three main columns. The left column of the interface is a control panel, where objects are arranged for the simulation control and interface. The middle column is an area for record of all important actions during the simulation. The right column is an area for displaying charts of all available values. The user can adjust number and type of displayed charts.

6.4 The data archiving

The application has the ability to save data to a file. The user can select the values wanted to store, and then download the results in the zip file after the simulation. The file contains a text in form of a matrix with data, where the columns are separated by commas and rows are separated by semicolons. The columns represent selected variables in the similar order to that in the user interface.

7 Simulator tests

7.1 The verification of simulator

The verification of the correct operation of the simulator and the quality of its results was made by comparing its simulation results with the reference simulator developed in the MATLAB / Simulink. The reference trajectory of the levels for the control rods of the reactor was prepared for both simulators. It assumed a step change in the set point position of control rods at defined levels of 175 cm, 225 cm, 200 cm and 150 cm changing in the sequence every 120 s. The results of the reference simulator have been saved directly into the MATLAB workspace. The simulation on the network simulator was initiated by one of the clients from the web browser and the results were saved to the file, which later also have been imported into the MATLAB workspace where the comparison with the reference simulator was made. The network simulator worked with the mechanism of step adaptation in the simulation with the initial step of the calculation set on 20 ms.

The reference simulator base step was determined to be 10 ms. Waveforms of neutron density were selected to the comparison. The resulting waveforms from both simulators rearranges fig.4.

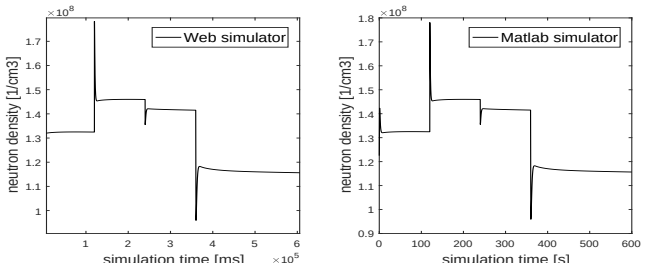

Fig. 4. The results of simulations - values of neutron density

The visual analysis of neutrons density waveforms did not show any significant deviations between the two simulators. The dynamics of the processes were recreated with the acceptable accuracy.

The direct numerical analysis of the results of both simulations was not possible due to the step-variable network simulator, which varied between 20 - 50 ms. To determine a quantitative measure of the quality of the simulation, the network simulator results were interpolated using the linear interpolation to match the number of the measurements of the reference simulator. The simulation errors were calculated using (1):

$$\sigma = \frac{\Sigma|\Delta X|}{\Sigma X} * 100\% \tag{1}$$

where: $\Delta X-$ the difference between the pair of results from the network and the reference simulator at the specific time, $X-$ the result of the reference simulator at the specific time

The calculated average error was 0.0064%. The largest relative errors could be observed at the time of a step change in the position of control rods. The maximal observed relative error of simulation results was 17.9602%, wherein it could be observed by only one step of the simulation. It is fully justified given the nature of the network simulator, the minor latency and the step change in the reference signal.

In the case of this simulator, in which changes in control signals do not occur too often, such errors have a marginal effect on the quality of the simulation. The resulting low average error proves the numerical correctness of the developed simulator.

7.2 The step length impact on a calculations quality

The quality tests in the network load conditions were performed. During the simulation repeatedly occurring delays were observed, resulting in frequent changes

of the simulation step length. The chart of neutrons density acquired during the test is presented in fig.5. Changes in the waveform of neutrons density not caused by the operation of the control system can be observed in the marked areas . The noise in waveforms can be seen especially in the area of the third minute of simulation.

Due to the impossibility of the stiffness elimination in the real-time simulator the numerical stability analysis with different simulation step lengths were performed. Increasing the simulation step, the extortion in the form of changing the position of the control rods and other factors were applied. Thus it was estimated that for the presented mathematical model and specified solver, numerically stable results can be obtained when the step length is not more than 50 ms.

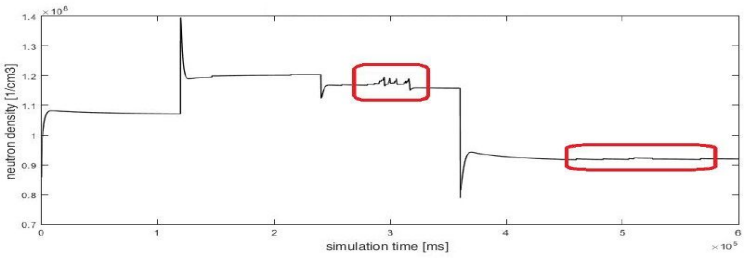

Fig. 5. The neutrons density waveform noise caused by aggressive simulation step changes under high load conditions

7.3 The web environment impact on a quality of simulation

The study involving the observation of the simulation step while communicating with different numbers of users was performed. The study began with four connected users. The application was tested with the maximum number of 15 clients. The negative impact on the communication time with the server was observed, what caused simulation delays when connecting with a larger number of clients at the same time. However, no impact on the step length of the continuously connected clients were found, because the server generates and sends the same version of the results for a group of connected clients in every cycle. In summary, the effect of the number of clients connected applications at the same time has a marginal impact on the quality of the simulation.

8 Summary and conclusions

The concept of the real-time basic principles nuclear reactor simulator based on the network environment is presented in the paper. The main considered problem was to achieve the numerically stable real-time simulation with the high

temporal resolution in the environment which is not a real-time for the plant with the extremely different time scales. The multi-step adaptation algorithm of the simulation step length was proposed for dealing with the real-time. The simulator program in the client-server architecture in the network environment was developed. Several studies were performed to test the developed simulator. The high compliance with the reference simulator was achieved and the maximum length of the step size was determined for ensuring the numerical stability. The only free and open software environments were applied. The user interface of the developed simulator is based on the web browser accessible on almost every operating system and computer hardware including the mobile. As a result of work undertaken in the article very useful education tool with the huge dissemination potential was developed.

References

1. *Nuclear Power Plant Simulator*, http://www.nuclearpowersimulator.com/, (access date 15.01.2017)
2. Rui Wu et al., *A Real-time Web-based Wildfire Simulation System*, IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 2016, pp. 4964-4969
3. Karla T., Tarnawski J., Duzinkiewicz K., *Cross-Platform Real-Time Nuclear Reactor Basic Principle Simulator*, 20th International Conference on Methods and Models in Automation and Robotics (MMAR), 2015, pp. 1074-1079. IEEE
4. Tarnawski J., Karla T., Rutkowski T.A., Puchalski B, Duzinkiewicz K., *Soft real-time simulation with adaptive step of computation*, The Scientific Papers of Faculty of Electrical and Control Engineering, Gdansk University of Technology, 2015
5. Tarnawski J., Karla T.,*Real-time simulation in non real-time environment*, 21st International Conference on Methods and Models in Automation and Robotics (MMAR), 2016, pp. 577-582
6. Popovici K., Mosterman P. J. (Eds.), *Real-time simulation technologies: principles, methodologies, and applications.*, CRC Press, 2012
7. Karla T., Tarnawski J. Duzinkiewicz K., *Symulator czasu rzeczywistego procesow reaktora jadrowego*, Aktualne Problemy Automatyki i Robotyki, Akademicka Oficyna Wydawnicza EXIT, 2014, pp. 558-569
8. Sokolski P. Rutkowski T.A. Duzinkiewicz K., *Simplified, multiregional fuzzy model of a nuclear power plant steam turbine.*, 21st International Conference on Methods and Models in Automation and Robotics (MMAR), 2016, pp. 379-384
9. Puchalski B., Rutkowski T.A., Duzinkiewicz K., *Multi-nodal PWR reactor model - methodology proposition for power distribution coefficients calculation.*, 21st International Conference on Methods and Models in Automation and Robotics (MMAR), pp. 385-390
10. Oka Y., Suzuki K., *Nuclear Reactor Kinetics and Plant Control*, Springer, 2013
11. Hetrick D. L., *Dynamics of Nuclear Reactors*, The University of Chicago Press, Chicago and London, 1971
12. Moler C., *Stiff Differential Equations*, Technical Articles and Newsletters, MathWorks, https://www.mathworks.com/company/newsletters/articles/stiff-differential-equations.html, (access date 15.01.2017)