

Hardware-Software Implementation of Basic Principles Simulator of Nuclear Reactor Processes

Authors

Tomasz Karla
Jarosław Tarnawski
Kazimierz Duzinkiewicz

Keywords

modeling, simulation, nuclear reactor, real-time requirements, education, training

Abstract

The paper presents implementation process of basic principle simulators of a nuclear reactor processes. Simulators are based on point-models of processes: kinetics of neutrons, heat generation and exchange, poisoning and burning-up nuclear fuel. Reference simulator was developed in MATLAB/Simulink without taking into account real-time operation. Second simulator was built using the toolbox xPC with hard real-time requirements. Functional decomposition of a nuclear reactor was performed and processes with different time scales were isolated. Simulation of each separated process in Raspberry PI computer with coordination mechanism among them led to a distributed soft real-time simulator. Idea of web-based simulator is also presented. The engine of the web-based simulator can be implemented on the server while the presentation of the state of the simulator and the prescription of parameters can be performed using a web browser. This allows simultaneous access to the simulator using different devices, including mobile as tablets, laptops and mobile phones. Compliance tests of simulators with reference simulator were conducted and the results proved the correctness of implementation. A review of the properties and potential applications of the various versions of developed simulators is presented.

DOI: 10.12736/issn.2300-3022.2016209

1. Introduction

Understanding the basic principles of nuclear reactor (NR) processes may be important for many reasons and for many potential recipients. In addition to theoretical knowledge simulators can be used to learn the dynamics of processes and their interdependencies as very valuable means of teaching. It is possible to specify various aspects of simulation, eg.: work in real time which reproduce time dependencies of processes that develop the sense of the dynamics of the object, allowing the testing of different scenarios with the analysis of the impact of individual parameters, the possibility of observing and archiving processes data or the ability to connect control systems for simulated object. For the users of simulators important aspects will be: the handling of simulator, user interface that allows one to manipulate the simulated object and observe the results of its operations. Above description presents that the different learning objectives and dissemination can be achieved with different versions of the simulators. The topics of simulators were already considered in [2, 3, 8, 9]. In this article authors have considered different hardware-software implementations of basic principle nuclear reactor simulators. Considered simulators are based on point models of processes and implemented using

Matlab/Simulink environment, real-time system xPC, Raspberry PI computer and network simulator operated via a web browser. In order to obtain real-time work authors considered distributed versions of simulators. This paper examines the possibility of obtaining step of the simulation corresponding to the actual speed of the neutron kinetic process, compliance simulators with the reference version. Paper contains also assessing the usefulness of simulators for different tasks, taking into account the most important criteria and utility costs.

2. Characteristics of the object dynamics

NR in simplified schemes is presented as a source of heat with regulated power [10]. The operation of the NR consists of a series of processes. One of the simplest but well reflecting the nature of the processes in the NR is point model, in which all variables are the values averaged over the volume of the core. More information about them can be found in [5, 6]. All simulators considered in this paper were based on such models [1]. Figure 1 presents processes and control effects of NR included in simulators.

First group includes nuclear reactions (neutron kinetics, poisoning and burnup of fuel) and heat transfer processes.

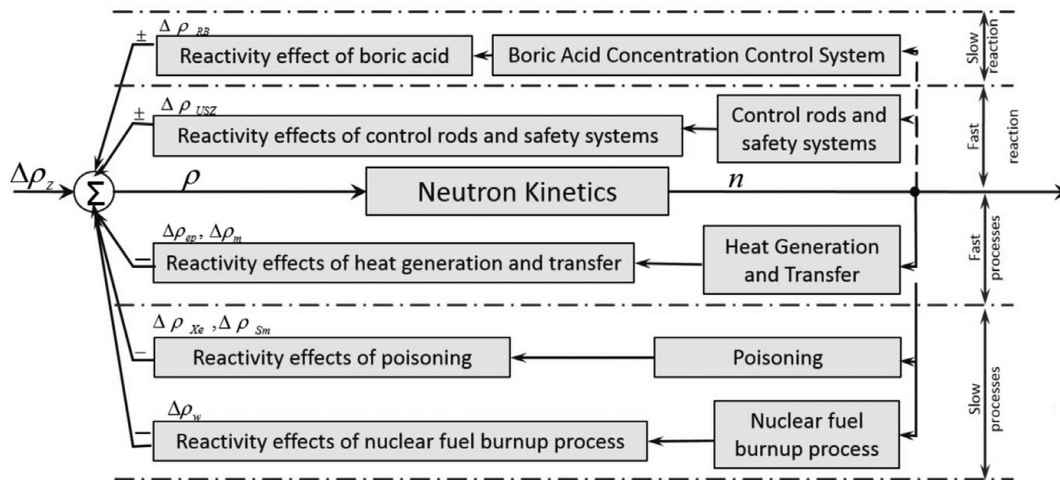


Fig. 1. The main processes occurring in a nuclear reactor affecting its state

Neutron kinetics includes changes of neutron density (n) in volume of the core depending on core's reactivity (ρ). Poisoning and burnup processes presents changes of reactor's reactivity ($\Delta\rho_{Xe,Sm}$) related with the formation of xenon ($Xe - 135$), iodine ($I - 135$), samarium ($Sm - 149$) and promethium ($Pm - 149$) in nuclear fuel as well as changes of reactivity ($\Delta\rho_w$) related to the change in the composition of the fuel during its burnup. Heat generation and transfer processes determine thermal conditions in the reactor related with neutron density and their changes of reactivity ($\Delta\rho_{ep,m}$). Control effects include operations related with control of reactivity through regulation of positions of control rods ($\Delta\rho_{RB}$) and boric acid concentration in moderator ($\Delta\rho_{USZ}$). All of these processes have different time scales. Table 1 summarizes the various processes and their transition times.

Proces	The duration of transient states
Neutron kinetics	10^{-5} to 10^{-3} seconds
Heat transfer between fuel and coolant/moderator, Changes in reactivity resulting from changes in temperatures of fuel and coolant/moderator	3 to 6 minutes
Change in reactivity by changing position of control rods	up to 125s
Change in reactivity by changing concentration of boric acid in moderator	several hours
Xenon poisoning occurring while changing power level of reactor	up to 60 hours
Samarium poisoning	15 to 20 days
Nuclear fuel burnup	tens of days and months

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

The difference between the fastest and the slowest processes occurring in the reactor reaches to several orders of magnitude. Therefore, they are further divided into quick and slow processes (Fig. 1).

3. Hardware and software realizations of simulators

There are many different possibilities of implementation of simulators, both in terms of hardware and software. Typical academic approach is application of popular software MATLAB/Simulink (M/S) running on PC class computers. Initially simulator in M/S was developed, and then variants of this reference simulator in other environments and hardware platforms were carried out: the simulator working in hard real-time with a small step of the simulation on PC, the simulator working in soft real-time on the hardware platform based on a low-cost device Raspberry PI (RPI) [11] and a variant of the network simulator that allows one to operate from a web browser. The distribution of simulation between multiple devices according to the ideas of multiagent operations [7] was considered to reduce the computation cost for the individual devices, and thus obtain a smaller step of the simulation.

3.1. Reference simulator MATLAB/Simulink

Basing on the mathematical model equations a simulation model in Simulink environment was developed. Solutions were obtained using solver ode3 based on Bogacki-Shampine algorithm. The calculations were performed in freerun mode (without time restrictions for calculations). The simulator provides a preview of all the process variables in the form of dedicated display blocks and/or in the form of charts waveforms, there is the ability to archive the results of simulations. The authors have adopted this version of the simulator as a reference because it does not have any simplifications, the simulation is performed in a centralized manner and performed independently of the real-time requirements allowing to get the simulation of any step.

3.2. Real-time simulators

Simulations performed in the reference simulator in the Simulink environment were not meant to work and present the results in real time. Depending on the adopted simulation step (accepted from range $1E-3s$ to $1E-5s$ corresponding to the average life time of neutrons) for calculating the results of simulation in specific moments, calculations are carried out about 10 times slower

to 100 times as fast as a real time. Work with the simulator that reflects dynamics of the real-time object has a large educational value, allow one to gain experience with the dynamics of the object which is necessary when working with real object. MATLAB/Simulink have toolboxes that allow a simulation while maintaining real-time operation. The authors have considered toolboxes Real-Time Windows Target (RTWT) and xPC, which allow one to run the simulation on PC and toolbox containing a package of modules to cooperate and compile applications on the platform Raspberry Pi.

3.2.1. Hard Real-time simulator with small step of simulation

Hard RT simulator that can work with the adopted by the authors the most challenging step of calculations at $1E-5s$ required the use of a solution having a high computing power. M/S with RTWT toolbox allows one to run simulations in RT mode in the Windows environment. However, the efficiency of this tool on test computers was insufficient at step $1E-5$. The alternative was toolbox xPC which is oriented on simulations in RT, offering much higher performance, because it performs calculations on specially prepared versions of the operating system with micro-kernel, containing only the necessary components needed

to carry out the simulation. In addition, another variant of the simulator was developed using the xPC, in which the simulation was distributed between 3 PCs. Time decomposition of nuclear reactor processes presented in Tab. 1 allowed to group the various processes in blocks, which can be performed with higher simulation steps (Fig. 2).

This solution required one to ensure communication between devices and the synchronization of each of the blocks of the simulation with the other blocks. The method of synchronization for moments of start, pause and end of calculations on individual devices was developed. Due to the limited visualization tools available in xPC toolbox, authors developed an additional application that runs on the fourth RTWT computer, which was used for the management of simulation, presentation of the results in real time and archiving. All units were connected to each other via a gigabit Ethernet switch using the connectionless UDP protocol (Fig. 3).

3.2.2. Soft Real-time simulator on Raspberry Pi platform

An alternative implementation involves using low-cost devices RPI to build the simulator running in real time with soft time constraints. Using Simulink toolbox Support Package for

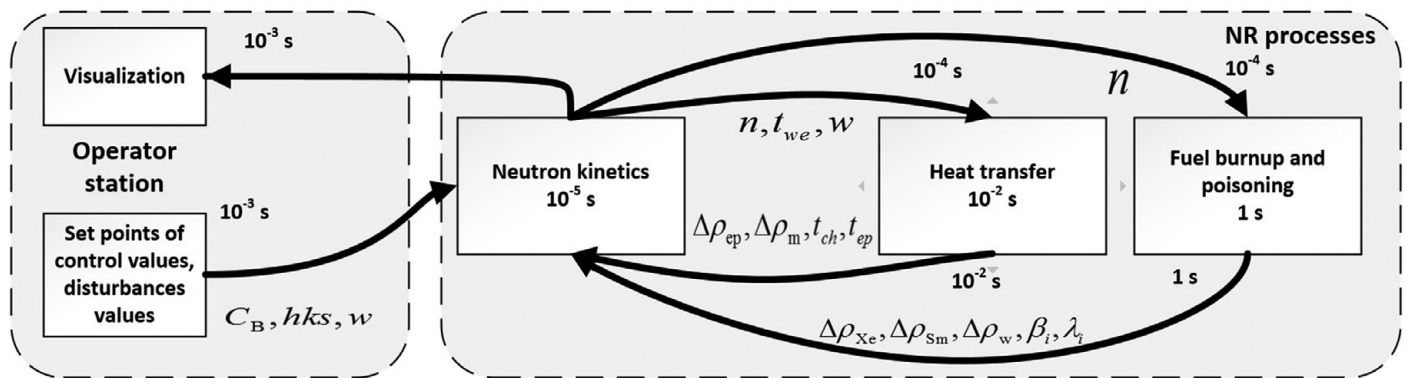


Fig. 2. The division into function and computing blocks

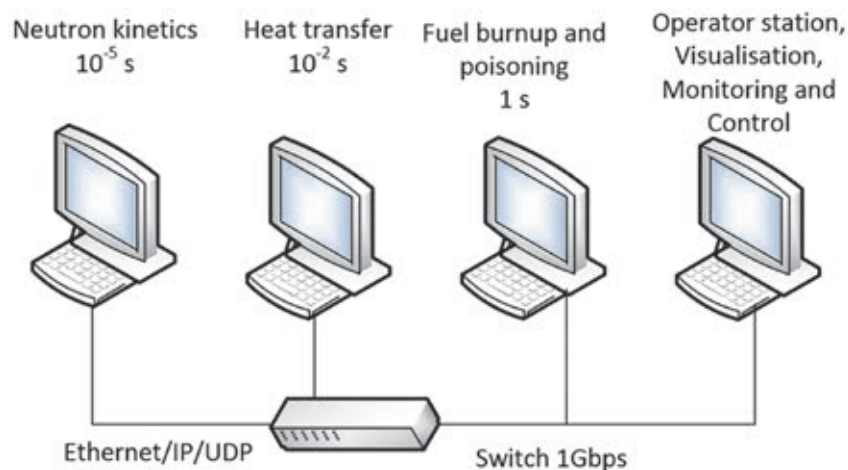


Fig. 3. Physical connections in the simulator

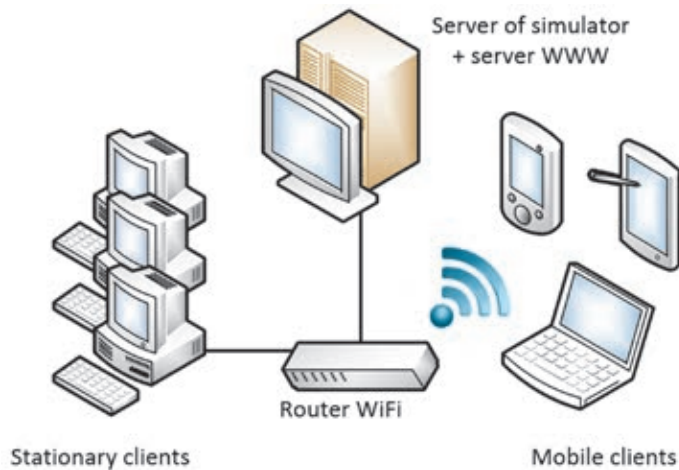


Fig. 4. Multi-access web-based simulator

Raspberry Pi RT simulator application was generated as a stand-alone application that runs without the use of M/S. The step of calculations was set at 1E-3 (the upper limit of the average life-time of neutrons). Toolbox allows the generation of application code but only if discrete blocks are used and allows one only to use the built-in solver called discrete. After the compilation application was executed on the designed platform RPI, but performance of a single device wasn't enough to assure the required step within reasonable constraints of soft RT for such a simulation. The solution was implementation of distributed simulation. Similarly as in the case of distributed simulation using the toolbox xPC variant simulator processes were divided into 3 blocks and implemented on three RPI. The mechanism of synchronization was implemented. The visualisation application which runs on the fourth RPI was developed in C code and was independent from M/S. Also in this case all units were connected to each other via a gigabit Ethernet switch using the connectionless UDP protocol. Further works assume direct implementation of simulator in C code, which can improve performance of the simulator on the RPI platform (reduce computational cost).

3.2.3. Multiple access Web-based simulator

The authors considered the implementation of a Web-based simulator in a server unit whose operations would be handled from a web browser. There isn't any simple solution that would guarantee its work in a hard time environment so the authors decided to implement it with soft real time requirements. Due to the applied infrastructure and its ability to provide multiple access to the simulator the implementation of the server must take into account the work of many instances. The target hardware platform is an efficient PC computer as a server-oriented simulator. The clients will be devices with web browsers that will have access to the server and can have a different character: classic PC, laptop, tablet, or even a mobile phone. This idea is presented in Fig. 4. Visualization can be designed using network and web technologies like: AJAX, JavaScript and HTML. The archiving of results can be handled by a PHP application.

3.3. Conformity assessment study of simulators

After implementation of RT simulators conformity assessment of their results was performed. The process of verification was divided into two parts. The first test concerned the correctness of communication between devices. Distributed versions of simulators performed calculations on different devices which could conclude with wrong results of simulation. Three aspects of communication were taken into consideration which could possibly have an effect on final results of calculations: packet losses, wrong order of received packets and latency in communication. All three aspects were examined using a mechanism developed by authors which uses additional numbering of consecutive sent packets. The loss of packets and wrong order was observed when numeration of received packets was discontinuous or changed. Latency was observed by counting time of transmission of packet marked with specified number from sending unit to target device which returns the number to sending unit again. After receiving an answer from the target device, in sending unit occurs comparison of send and receive times of specified packet number and latency time is calculated. The second test was to verify the simulation results of reference and RT based simulators which run the same pre-made scenarios of control and disturbances trajectories included in the model. By comparing the results of the reference and RT simulators at certain moments of time the average simulation error determined by the formula (1) for each of the RT simulators was calculated. The results of the verification process are presented in Tab. 2.

$$e_w = \frac{1}{N} \sum_{i=0}^N \left(\frac{w_r(i) - w_s(i)}{w_r(i)} \right)^2 * 100\% \quad (1)$$

where:

$w_r(i)$ – value from the reference simulator, $w_s(i)$ – value from RT simulator, N – number of simulation samples.

4. Summary of the characteristics of developed simulators

The presented simulators have different features which allow assessment of their suitability for potential applications. The authors adopted some criteria of quality, in terms of evaluating the various simulators. One of them is the Conformity, determined on the basis of conformity between the results of simulation described in Section 3.3. Another criterion is the ability to work in RT mode, which greatly increases the feeling of working with a real object by simulating its actual dynamics. The criterion

	Packet loss	Wrong order of packets	Average latency	Average error
xPC	0.00003%	0%	0.8ms	0.3%
RPI	0.00016%	0%	7ms	0.65%

Tab. 2. Summary of the results of the verification process of simulators

	Conformity	Real-Time	Operator options	Archiving	Scenarios	HMI	HIL	Dissemination	Cost
MATLAB/Simulink	Reference	Absent	Standard controls of M/S	High	High	Medium	Absent	Medium	Medium
xPC	High	Hard	Simple, Limited, Difficult	Medium	Medium	Medium	Medium	Low	High
RPI	Medium	Soft/Hard	Simplified – application designed by authors for RPI	Medium	Medium	High	High	Very High	Medium/Low
Web-based	Medium	Soft	Flexible, convenient, web technology controls	High	Medium	High	Absent	High	Low

Tab. 3. Summary of features of presented simulators

for determining range of the possibilities of influencing the parameters and variables of the object is specified as the Operator options. An important feature is the possibility of archiving simulation results for later analysis. Criterion Scenarios defines the possibilities of programming of certain trajectories of control and disturbances variables in the simulation runs. Possibilities of visualisation of processes which are commonly known as user interface or HMI (Human Machine Interface) were assessed. RT simulators can be used in Hardware in the Loop (described in [4]) which was taken into consideration when comparing features of simulators. Availability of solutions, hardware and software, as well as the popularity of the tools were included in the criterion Dissemination. The last criterion taken into consideration is the cost of simulator implementation. The features of simulators incorporating all the criteria are listed in Tab. 3.

All implementations provide simulation results of satisfactory quality and sufficient for the purposes of training and education. RT operation mode is possible in versions of the xPC and RPI and web-based simulator, but due to the variation in terms of computing power available to work with hard real-time constraints with smaller step of simulation is only possible for the xPC. There is also a significant difference in terms of archiving. Reference simulator M/S and the web-based one can save the results of all process variables directly from the simulation. In case of xPC and RPI archiving is done by visualization applications, so it is possible to only record the observed values. The greatest potential for the development of work scenarios has reference simulator M/S, because it is possible to freely modify all the variables and structures directly in the model. The possibilities of other simulators in this aspect are dependent directly on implemented mechanisms of visualization applications. Reference simulator uses blocks display and scope available in the M/S. Visualisation Applications have clear graphics and are ergonomically designed, depending on the hardware platform. Taking into account the costs of simulators xPC and M/S versions find potential use in academic units while popularizing actions

will be more appropriate for RPI low-budget projects and a web-based simulator.

5. Conclusion

The paper presents and compares different software and hardware implementations of basic principles of nuclear reactor processes simulators. On the basis of mathematical point models of reactor processes a reference simulator was built in MATLAB/Simulink, which may be used in a wide range of research and analysis. Two another simulators were created that operate in real time with hard and soft time requirements. The results of the compliance test showed only minor deviations to the reference simulator so it makes them well suited for research requiring real-time performance, eg. Hardware-in-the-Loop. In addition, they allow one to get a better sense of the dynamics of the real object than the reference version of the simulator which also has a high educational value. The authors also present the concept of a web-based simulator, which is the subject of further work. Functional analysis of all simulators taking into account multi-various criteria, such as operator capabilities, availability and costs, allowed determining their applications between typically academic and dissemination.

REFERENCES

1. G. Baum et al., "Modele symulacyjne procesów zachodzących w podstawowych urządzeniach obiegu pierwotnego w warunkach normalnej eksploatacji i niedużych zakłóceń (SYMREP)", Instytut Elektroenergetyki i Automatyki Politechniki Gdańskiej, 1989.
2. T. Karla, J Tarnawski. K Duzinkiewicz, "Symulator czasu rzeczywistego procesów reaktora jądrowego", Aktualne Problemy Automatyki i Robotyki, Akademicka Oficyna Wydawnicza EXIT, 2014, pp. 558–569
3. K. Duzinkiewicz, A. Cimiński, Ł. Michalczyk, "Symulator szybkich procesów dynamiki reaktorajądrowego wodnego ciśnieniowego", Pomiary Automatyka Robotyka, No. 9, 2013, pp. 97–101.

4. B. Puchalski et al., "Programowo-sprzętowa platforma symulacyjna – Hardware In the Loop – zaawansowanego układu sterowania poziomem wody w pionowej wytownicy pary elektrowni jądrowej", Aktualne Problemy Automatyki i Robotyki, Akademicka Oficyna Wydawnicza EXIT, 2014, pp. 570–580.
5. Y. Oka, K. Suzuki, "Nuclear Reactor Kinetics and Plant Control", Springer 2013.
6. D.L. Hetrick, "Dynamics of Nuclear Reactors", The University of Chicago Press, Chicago and London, 1971.
7. L. Sterling, K. Taveter, "The Art of Agent-Oriented Modeling", The MIT Press, 2009.
8. H. Seok et al., "Development of software for the microsimulator for the KO-RI nuclear power plant unit 2", Nuclear technology, No. 106(3), 1994, pp. 384–396.
9. A. Kluge, et al., "Designing training for process control simulators: a review of empirical findings and current practices", Theoretical Issues in Ergonomics Science, No. 10(6), 2009, pp. 489–509.
10. D. Flynn (Ed.), "Thermal power plant simulation and control", IET, No. 43, 2003.
11. Raspberry PI Website <http://www.raspberrypi.org>, Accessed on 30.10.2014.

Tomasz Karla

Gdańsk University of Technology

e-mail: tomasz.karla@pg.gda.pl

In 2013 received the degree of MSc. at the Gdańsk University of Technology, Faculty of Electrical and Control Engineering and then began doctoral studies in a multi-agent control applied to the primary circuit of a nuclear power plant reactor PWR. Participant in the study of Innovative Nuclear Power and Sustainable Engineering at the Warsaw University of Technology.

Jarosław Tarnawski

Gdańsk University of Technology

e-mail: jaroslaw.tarnawski@pg.gda.pl

In 2000 he received the degree of MSc. at the Gdańsk University of Technology, Faculty of Electrical and Control Engineering, and then began doctoral studies in the subject of applications of adaptive control to quality of water in drinking water distribution networks. He received PhD in 2006. Research areas are implementation of advanced control methods, decentralization of control, multi-agent systems.

Kazimierz Duzinkiewicz

Gdańsk University of Technology

e-mail: kazimierz.duzinkiewicz@pg.gda.pl

He received MSc degree in Electrical Engineering and PhD in Control Engineering from Faculty of Electrical and Control Engineering at the Gdańsk University of Technology, in 1973 and 1982, respectively. In 2009 he received the Dr hab. Eng. degree in Control Engineering from Faculty of Electrical Engineering, Control Engineering, Informatics and Electronic Engineering at the AGH University of Science and Technology. He has been employed as a university teacher starting his work in 1973 from the post of Assistant to the current position of Associate Professor in the Department of Control Engineering. Now he is the Head of the Department of Control Systems Engineering. During his research work he has published over 100 reviewed scientific papers and 50 scientific and technical reports, mainly dealing with the following problems: a) production scheduling and operational control of technological systems with switchable processes, b) computer control of electric power station in emergency conditions, c) safety and reliability analysis of hazardous systems, d) mathematical modelling of complex systems, e) multihorizon and multilevel optimisation and control structures and algorithms.

This is a supporting translation of the original text published in this issue of "Acta Energetica" on pages 107–112. When referring to the article please refer to the original text.

PL

Realizacje sprzętowo-programowe symulatorów podstawowych zasad przebiegu procesów reaktora jądrowego

Autorzy

Tomasz Karla
Jarosław Tarnawski
Kazimierz Duzinkiewicz

Słowa kluczowe

symulatory, reaktor jądrowy, kształcenie, praca w czasie rzeczywistym

Streszczenie

W artykule przedstawiono możliwości i proces budowy symulatorów podstawowych zasad działania procesów reaktora jądrowego. Proces budowy i badania oparte są na modelach punktowych: kinetyki neutronów, procesów generacji i wymiany ciepła oraz procesów zatrzymania i wypalania paliwa. Badano możliwości realizacji symulatorów, wykorzystując różne środowiska programowe i platformy sprzętowe. Referencyjny symulator został opracowany w środowisku MATLAB/Simulink bez uwzględnienia wymagań pracy w czasie rzeczywistym. Na jego podstawie zbudowano w środowisku xPC drugi symulator procesów reaktora, pracujący w czasie rzeczywistym przy spełnieniu twardych wymagań czasowych. Następnie dokonano dekompozycji funkcjonalnej reaktora jądrowego i wydzielono procesy o różnych skalach czasowych, a ich symulacje zostały wraz z mechanizmem synchronizacji zaimplementowane w kilku jednostkach Raspberry PI, uzyskując rozproszoną wersję symulatora. Przeprowadzono weryfikację poprawności działania symulatorów oraz opracowano interfejsy użytkownika i instrukcje obsługi. W artykule jest również rozpatrywany sieciowy symulator obsługiwany z poziomu przeglądarki internetowej. Dokonano porównania funkcjonalności, ograniczeń i potencjalnych zastosowań poszczególnych symulatorów.

1. Wprowadzenie

Rozumienie podstawowych zasad przebiegu procesów reaktora jądrowego może być istotne z wielu powodów i dla wielu potencjalnych odbiorców. Oprócz wiedzy teoretycznej wartościowym środkiem dydaktycznym mogą być symulatory umożliwiające zapoznanie się z dynamiką procesów i ich współzależnościami. Można wskazać rozmaite aspekty symulacji, np.: pracę w czasie rzeczywistym, czyli odwzorowanie zależności czasowych w celu wyrobienia poczucia dynamiki obiektu, umożliwienie testowania różnych scenariuszy z analizą wpływu poszczególnych parametrów, możliwość obserwacji i rejestracji symulowanych wielkości czy możliwość podłączenia układów sterowania do symulowanego obiektu. Dla użytkowników symulatorów istotne będą: metoda obsługi, rodzaj sprzętu i oprogramowania czyli interfejs użytkownika, który umożliwi wpływ na symulowany obiekt i obserwację wyników jego działania. Z powyższego opisu wynika, że różne cele dydaktyczne i popularyzatorskie można osiągnąć z różnymi wersjami symulatorów. Tematykę symulatorów podejmowano już w literaturze, m.in. [2, 3, 8, 9]. W artykule są rozpatrywane różne implementacje sprzętowo-programowe symulatorów podstawowych zasad przebiegu procesów reaktora jądrowego. Na podstawie modeli punktowych procesów rozpatrywane są symulatory realizowane z wykorzystaniem pakietu Matlab/Simulink, systemu czasu rzeczywistego xPC, komputerów Raspberry PI oraz symulator sieciowy obsługiwany za pomocą przeglądarki internetowej. Na potrzeby uzyskania pracy w czasie rzeczywistym rozpatrywane są wersje rozproszone symulatorów. W artykule badane są: możliwość uzyskania kroku symulacji odpowiadającego rzeczywistej prędkości procesu kinetyki neutronów, zgodność symulatorów z wersją referencyjną. Oceniana jest przydatność symulatorów do różnych zadań, biorąc

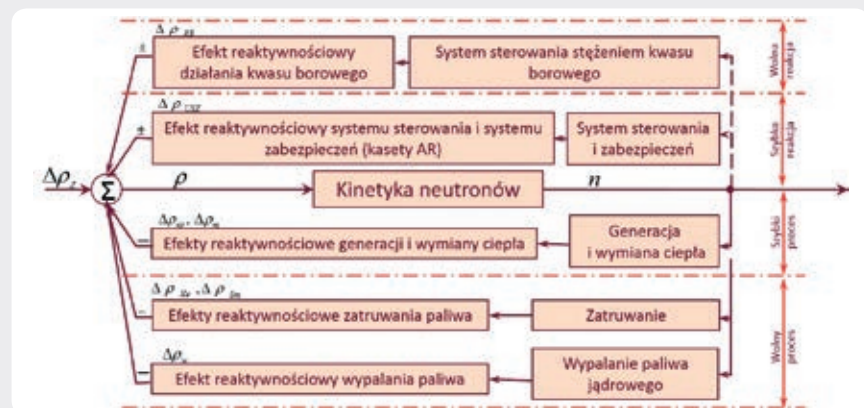
pod uwagę najważniejsze użytkowe kryteria, w tym koszty.

2. Charakterystyka dynamiki obiektu

Reaktor jądrowy (RJ) w uproszczonych schematach elektroni jest przedstawiany jako źródło ciepła o regulowanej wartości mocy [10]. Na pracę reaktora jądrowego składa się wiele procesów. Do ich zamodelowania można użyć modeli o różnej złożoności. Ich dokładny opis można znaleźć w literaturze [5, 6]. Najprostszym z nich, jednak dobrze oddającym charakter procesów w RJ, jest model punktowy, w którym zmiennymi są wielkości uśrednione po objętości rdzenia. Wszystkie rozpatrywane w artykule symulatory zostały zatem oparte na modelach punktowych [1]. Uwzględnione w symulatorach procesy i oddziaływania sterowania reaktorem zostały przedstawione na rys. 1. Do pierwszej grupy zaliczają się reakcje jądrowe (kinetyka neutronów, zatrzymanie i wypalanie paliwa) i procesy wymiany ciepła. Kinetyka neutronów uwzględni

zmiany koncentracji neutronów (n) w objętości rdzenia, w zależności od reaktywności rdzenia (ρ). Procesy zatrzymania i wypalania paliwa przedstawiają zmiany reaktywności ($\Delta\rho_{Xe,sm}$) reaktora, związane z powstawaniem ksenonu ($Xe-135$), jodu ($I-135$), samaru ($Sm-149$) i prometu ($Pm-149$) w paliwie, oraz zmiany reaktywności ($\Delta\rho_w$) powiązane ze zmianą składu samego paliwa w czasie jego zużywania. Procesy generacji i wymiany ciepła determinują warunki cieplne w reaktorze w zależności od koncentracji neutronów i ich wpływ na zmiany reaktywności ($\Delta\rho_{ep,m}$).

Do wymuszeń sterowniczych należą operacje związane ze sterowaniem reaktywnością reaktora poprzez wpływanie na pozycję prętów sterujących ($\Delta\rho_{RB}$) w rdzeniu oraz na stężenie kwasu borowego w moderatorze ($\Delta\rho_{USZ}$). Wszystkie te procesy różnią się skalami czasowymi, w jakich zachodzą. Tabela 1 przedstawia zestawienie poszczególnych procesów i czasów trwania ich procesów przejściowych.



Rys. 1. Główne procesy zachodzące w reaktorze jądrowym, wpływające na jego stan

Różnice między najwolniejszymi i najszybszymi procesami sięgają kilku lub kilkunastu rzędów wielkości. Dlatego też dzieli się je dodatkowo na procesy szybkie i wolne (rys. 1).

3. Realizacje sprzętowo-programowe symulatorów

Istnieje wiele różnych możliwości implementacji symulatorów zarówno pod względem sprzętowym, jak i programowym. Typowym, bazowym podejściem akademickim jest wykorzystanie popularnego oprogramowania MATLAB/Simulink (M/S), uruchamianego na komputerach klasy PC. Przyjęty plan prac nad symulatorami zakładał przygotowanie początkowo symulatora w tym właśnie środowisku, a następnie w oparciu o ten referencyjny symulator próby implementacji jego odmian w innych środowiskach i na innych platformach sprzętowych. Zaliczono do nich: symulator pracujący w czasie rzeczywistym (Real-Time RT), o twardych wymaganiach czasowych, z małym krokiem symulacji na platformie komputerów klasy PC, symulator pracujący w czasie rzeczywistym o miękkich, wymaganiach czasowych, na platformie sprzętowej opartej na niskobudżetowych urządzeniach Raspberry PI (RPI) [11] oraz wariant symulatora sieciowego, który pozwala na obsługę z poziomu przeglądarki internetowej. Uwzględniono rozproszenie symulacji między wiele urządzeń, aby zmniejszyć nakład obliczeń na pojedynczych urządzeniach i tym samym uzyskać mniejszy krok symulacji wg idei wieloagentowości [7].

3.1. Referencyjny symulator MATLAB/Simulink

Na podstawie równań modelu matematycznego opracowano model symulacyjny w środowisku Simulink. Do rozwiązywania obliczeń wykorzystano solver *ode3* oparty na algorytmie Bogackiego-Shampine'a. Obliczenia były wykonywane w trybie *freerun* (bez stawiania wymagań czasowych dla obliczeń). Symulator udostępnia podgląd wszystkich wielkości procesowych w postaci dedykowanych blozków wyświetlaczy i/lub w postaci wykresów przebiegów, istnieje też możliwość archiwizacji wyników symulacji. Autorzy przyjęli tę wersję symulatora jako referencyjną, ponieważ nie posiada ona żadnych uproszczeń, symulacja jest wykonywana w sposób scentralizowany i wykonywana niezależnie od wymagań czasu rzeczywistego, co umożliwia otrzymanie symulacji z dowolnym krokiem.

Proces	Czasy procesów przejściowych
Kinetyka neutronowa	10 ⁻⁵ –10 ⁻³ s
Procesy wymiany ciepła między paliwem a chłodziwem/moderatorem, zmiana reaktywności związana ze zmianami temperatury	3–6 min
Zmiana reaktywności za pomocą regulacji poziomu zanurzenia prętów sterujących	do 125 s
Zmiana reaktywności za pomocą regulacji stężenia kwasu borowego	kilka godzin
Zatrucie ksenonem w czasie zmiany mocy reaktora	do 60 godz.
Zatrucie samarem	15–20 dni
Wypalanie paliwa	dziesiątki dni/miesiące

Tab. 1. Przedziały czasowe poszczególnych procesów zachodzących w reaktorze

3.2. Symulatory RT

Symulacje wykonywane w referencyjnym symulatorze w środowisku Simulink nie były przeznaczone do pracy i prezentacji wyników w czasie rzeczywistym. W zależności od przyjętego kroku symulacji (przyjętego z zakresu 1E-3s do 1E-5s odpowiadającego średniemu czasowi życia neutronów), obliczenia dla danych chwil czasowych były wykonywane od ok. 10 razy wolniej do 100 razy szybciej w stosunku do czasu rzeczywistego. Praca z symulatorem odzwierciedlającym dynamikę obiektu w czasie rzeczywistym posiada duże walory edukacyjne, pozwala nabrać doświadczeń związanych z dynamiką obiektu konieczną przy jego obsłudze. Środowisko MATLAB/Simulink posiada przyborki pozwalające na wykonywanie symulacji z zachowaniem pracy w czasie rzeczywistym. Autorzy rozpatrywali przyborki Real-Time Windows Target (RTWT) i xPC, pozwalające na symulację na komputerach klasy PC oraz przyborkę zawierającą pakiet modułów do współpracy i kompilowania aplikacji na platformę RPI.

3.2.1. Symulator RT o twardych ograniczeniach czasowych i małym kroku symulacji

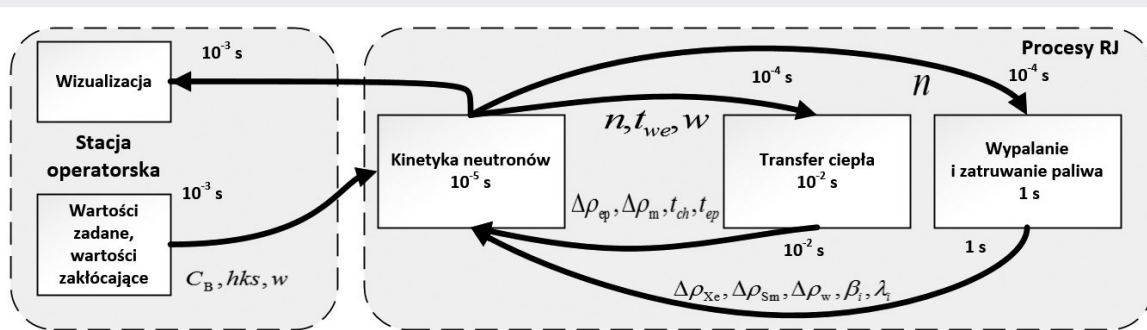
Zbudowanie symulatora RT, który mógł pracować z przyjętym przez autorów najbardziej wymagającym krokiem obliczeń rzędu 1E-5s, wymagało zastosowania rozwiązania mającego duże moce obliczeniowe. M/S za pomocą przyborki RTWT pozwala na przeprowadzanie symulacji w trybie RT w środowisku Windows. Jednak wydajność tego narzędzia na komputerach testowych była niewystarczająca przy kroku 1E-5. Alternatywą był przyborek xPC,

zorientowany na symulację w trybie RT, oferujący znacznie wyższą wydajność, gdyż realizuje obliczenia na specjalnie przygotowanych wersjach systemu operacyjnego z mikrojądrem, zawierającym tylko niezbędne komponenty potrzebne do przeprowadzania symulacji. Dodatkowo opracowano inny wariant symulatora z wykorzystaniem xPC, w którym dokonano rozproszenia symulacji pomiędzy trzy komputery klasy PC. Dekompozycja czasowa procesów reaktora jądrowego przedstawiona w tab. 1 pozwoliła na pogrupowanie poszczególnych procesów w bloki, które mogły być wykonywane z większymi krokami obliczeń (rys. 2).

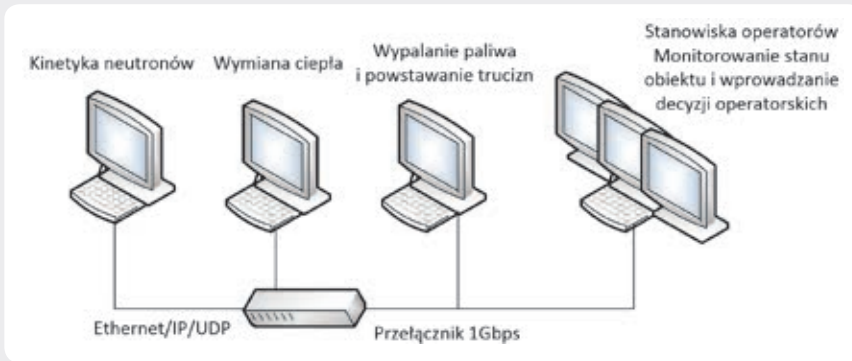
Rozwiązanie takie wymagało jednak zapewnienia komunikacji między poszczególnymi urządzeniami oraz synchronizacji każdego z bloków symulacji z pozostałymi. Opracowano metodę synchronizacji momentów rozpoczęcia, wstrzymania i końca obliczeń na poszczególnych urządzeniach. Ze względu na ograniczone możliwości wizualizacyjne narzędzia, opracowano dodatkową aplikację za pomocą narzędzia RTWT, uruchamianą na czwartym komputerze, która służyła do zarządzania symulacją, prezentacji wyników w czasie rzeczywistym i ich archiwizacji. Wszystkie jednostki były połączone ze sobą za pomocą gigabitowego przełącznika Ethernet, przy wykorzystaniu bezpołączeniowego protokołu UDP (rys. 3).

3.2.2. Symulator RT o miękkich ograniczeniach czasowych w RPI

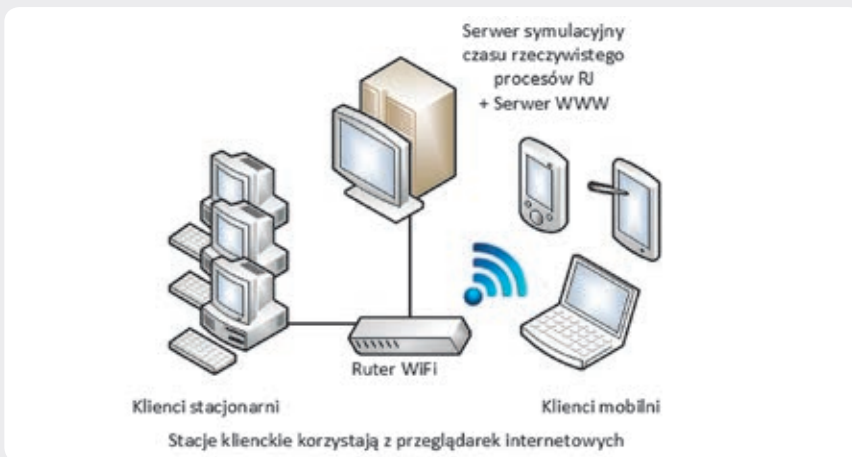
Alternatywna implementacja zakłada wykorzystanie niskobudżetowych urządzeń RPI do budowy symulatora pracującego



Rys. 2. Podział na bloki funkcyjne i obliczeniowe



Rys. 3. Fizyczne połączenia w symulatorze



Rys. 4. Wielodostępowy koordynowany symulator quasi-czasu rzeczywistego

	Utrata pakietów	Zła kolejność pakietów	Średnie opóźnienia	Średni błąd
xPC	0,00003%	0%	0,8 ms	0,3%
RPI	0,00016%	0%	7 ms	0,65%

Tab. 2. Zestawienie wyników procesu weryfikacji symulatorów

w czasie rzeczywistym z miękkimi ograniczeniami czasowymi. Za pomocą przybornika *Simulink Support Package for Raspberry PI* wygenerowano aplikację symulatora RT typu *stand-alone*, uruchamianą bez użycia M/S. Jako docelowy krok obliczeń przyjęto 1E-3s (górny próg ze średniego czasu życia neutronów). Przybornik pozwalał na generację kodu aplikacji tylko w wypadku używania bloków dyskretnych i użycia wbudowanego solvera o nazwie *discrete*. Po kompilacji aplikacji i uruchomieniu na docelowej platformie nie udało się uzyskać wymaganego kroku obliczeń, wykorzystując jedno urządzenie. Dokonano więc, wzorem wersji rozproszonej xPC, podziału procesów i poszczególne bloki zaimplementowano łącznie na 3 RPI. Analogicznie jak w wypadku rozproszonej symulacji przy użyciu przybornika xPC, tak i w tym wypadku użyto opracowanych mechanizmów synchronizacji symulacji na poszczególnych urządzeniach. Opracowano także w języku C niezależną od M/S aplikację wizualizacyjną, która mogła być uruchamiania na czwartym RPI. Podobnie jak w wypadku

rozwiązania opartego na narzędziu xPC, tak i w tym wypadku do komunikacji między wszystkimi urządzeniami użyto połączenia przez interfejs Ethernet, z wykorzystaniem gigabitowego przełącznika i protokołu UDP. Dalsze prace przewidują niezależenie kodu aplikacji symulatora od środowiska M/S i jego implementację w języku C, co ma także na celu dalsze poprawienie wydajności symulatora (osiągnięcie jak najmniejszego kroku obliczeniowego).

3.2.3. Wielodostępowy symulator sieciowy (Web-based)

Autorzy rozpatrują zbudowanie sieciowego symulatora zrealizowanego na serwerze i obsługiwanego z przeglądarki internetowej. Wobec braku prostej możliwości otrzymania symulacji z gwarancją czasu rzeczywistego przewiduje się symulator RT o miękkich wymaganiach czasowych. Ze względu na zastosowaną infrastrukturę i jej możliwości przewiduje się wielodostęp do symulatora, zatem realizacja na serwerze musi uwzględniać pracę wielu instancji. Docelową platformą sprzętową zostanie wydajny

komputer klasy PC, zorientowany na rolę serwera symulatora. Rolę klientów zaś będą pełniły urządzenia z przeglądarkami internetowymi, które będą miały dostęp do serwera i mogą mieć różny charakter: klasycznego peceta, laptopa, tabletu czy nawet telefonu komórkowego. Idea tego rozwiązania jest zaprezentowana na rys. 4. Strona prezentacyjna tej wersji symulatora może być opracowana z wykorzystaniem technik sieciowych, takich jak: AJAX, JavaScript i HTML, natomiast do archiwizacji może posłużyć mechanizm oparty na PHP.

3.3. Badania oceny zgodności symulatorów

Po implementacji symulatorów RT dokonano weryfikacji poprawności ich wyników. Weryfikacja była dwuetapowa. Pierwszy etap polegał na sprawdzeniu poprawności komunikacji między urządzeniami. Rozproszone odmiany symulatorów RT dokonywały symulacji w różnych urządzeniach, co potencjalnie może prowadzić do błędnych wyników symulacji. Wzięto pod uwagę trzy aspekty komunikacji, które mogą wpływać na błędy w wynikach symulacji: utrata pakietów, docieranie pakietów w nieprawidłowej kolejności oraz opóźnienia w transmisji. Wszystkie trzy aspekty zostały przebadane za pomocą opracowanego mechanizmu, wykorzystującego dodatkową numerację kolejnych pakietów. Utrata pakietów oraz docieranie w złej kolejności było obserwowane, gdy numeracja była nieciągła lub zamieniona. Opóźnienia obserwowano, zliczając czas transmisji pakietu o danym numerze do jednostki, która zaraz po jego otrzymaniu zwracała jego numer do jednostki, z której go otrzymała. Po uzyskaniu odpowiedzi, w jednostce wysyłającej porównane zostają czasy wysłania i ponownego odebrania danego numeru pakietu. Na podstawie tej różnicy były obliczane opóźnienia w transmisji. Drugi etap weryfikacji polegał na porównaniu wyników symulatora referencyjnego oraz RT na podstawie tych samych, wcześniej przygotowanych scenariuszy przebiegów zmiennych sterujących i zakłóceń uwzględnionych w użytym modelu procesów reaktora jądrowego. Porównując wyniki poszczególnych symulatorów w określonych chwilach czasowych określano wg wzoru (1) średni błąd symulacji dla każdego z symulatorów RT. Wyniki procesu weryfikacji przedstawia tab. 2.

$$e_w = \frac{1}{N} \sum_{i=0}^N \sqrt{\left(\frac{w_r(i) - w_s(i)}{w_r(i)} \right)^2} * 100\% \quad (1)$$

gdzie:

$w_r(i)$ – wartość referencyjna z symulatora niepracującego w RT, $w_s(i)$ – pomiar z symulatora RT, N – liczba pomiarów.

4. Zestawienie cech opracowanych symulatorów

Przedstawione symulatory posiadają różne cechy, które umożliwiają ocenę ich przydatności do potencjalnych zastosowań. Przyjęto kilka kryteriów jakościowych, według których oceniano poszczególne symulatory. Jednym z nich jest „Zgodność”, określana na podstawie zgodności wyników symulacji opisanej w rozdziale 3.3. Kolejnym kryterium jest możliwość pracy w trybie RT,

	Zgodność	Real-Time	Opcje operatorskie	Archiwizacja	Scenariusze	HMI	HIL	Rozpowszechnienie	Koszt
MATLAB/Simulink	Wzór	Brak	Oparte na kontrolkach dostępnych w M/S	Wysoki	Wysoki	Średni	Brak	Średni	Średni
xPC	Wysoka	Twardy	Proste, ograniczone, utrudnione	Średni	Średni	Średni	Średni	Niski	Wysoki
RPI	Średnia	Twardy/Miękki	Uproszczone – autorska aplikacja dla RPI	Średni	Średni	Wysoki	Wysoki	Bardzo wysoki	Średni/Niski
Sieciowy	Średnia	Miękki	Elastyczne, wygodne, oparte na kontrolkach przeglądarki	Wysoki	Średni	Wysoki	Brak	Wysoki	Niski

Tab. 3. Zestawienie cech przedstawionych symulatorów

co znacznie podwyższa poczucie pracy z rzeczywistym obiektem, oddając jego rzeczywistą dynamikę. Kryterium określające rozbudowanie możliwości wpływania na parametry i zmienne obiektu określono jako „Opcje operatorskie”. Ważną cechą jest też możliwość „Archiwizacji” przebiegów symulacji do późniejszych analiz. Kryterium „Scenariusze” określa możliwość programowania określonych przebiegów zmiennych w symulacji. Oceniano też możliwości prezentacyjne symulatorów, tj. ogólnie pojęty interfejs użytkownika. Symulatory RT obiektów mogą być wykorzystywane w pętach sprzężenia zwrotnego (ang. *Hardware in the Loop* – HIL) opisanych w [4], co także wzięto pod uwagę. Dostępność rozwiązań, urządzeń i oprogramowania oraz popularność narzędzi zawarto pod kryterium „Rozpowszechnienie”. Ostatnim wziętym pod uwagę kryterium jest „Koszt” implementacji symulatora. Zestawienia cech symulatorów uwzględniające wszystkie wymienione kryteria dokonano w tab. 3.

Wszystkie implementacje zapewniają zadowalającą jakość wyników symulacji, wystarczającą do celów szkoleniowych i edukacyjnych. Praca w trybie RT jest możliwa w wersjach xPC oraz RPI oraz w wariancie sieciowym, przy czym ze względu na zróżnicowanie pod względem dostępnych mocy obliczeniowych praca z twardymi ograniczeniami czasu rzeczywistego, przy mniejszych krokach symulacji, możliwa jest tylko na xPC. Występuje także zróżnicowanie pod względem możliwości archiwizacji. Symulator referencyjny M/S oraz sieciowy mogą zapisywać wyniki wszystkich zmiennych procesowych symulacji, bezpośrednio mając do nich dostęp. Archiwizacja w wypadku xPC oraz RPI odbywa się poprzez aplikacje wizualizacji, więc możliwy jest zapis przebiegu tylko obserwowanych wartości. Największe możliwości kształtowania scenariuszy pracy posiada symulator referencyjny M/S, gdyż można dowolnie modyfikować wszystkie zmienne i struktury bezpośrednio w modelu. Możliwości pozostałych symulatorów w tym aspekcie są powiązane bezpośrednio z zaimplementowanymi w aplikacjach wizualizacji mechanizmami. Symulator referencyjny używa

dostępnych w M/S bloków prezentacji wykresów i wartości. Aplikacje wizualizacyjne mają czytelne oprawy graficzne i są ergonomiczne, uzależnione od platformy sprzętowej. Biorąc pod uwagę koszty realizacji symulatorów, wersje xPC oraz M/S znajdują potencjalne zastosowanie w jednostkach akademickich, natomiast do działań popularyzatorskich bardziej odpowiednie będą realizacje niskobudżetowe RPI oraz symulator sieciowy.

5. Podsumowanie

W artykule przedstawiono i porównano różne implementacje programowo-sprzętowe symulatorów podstawowych zasad przebiegów procesów reaktora jądrowego. Na podstawie punktowych modeli matematycznych procesów reaktora zbudowano referencyjny symulator w środowisku MATLAB/Simulink, który może być zastosowany w szerokim zakresie badań i analiz. Następnie, bazując na nim, utworzono dwa symulatory pracujące w czasie rzeczywistym, z odpowiednio twardymi i miękkimi wymaganiami czasowymi. Wyniki badań testowych wykazały tylko niewielkie odstępstwa wyników symulacji tych symulatorów względem symulatora referencyjnego. Sprawia to, że dobrze nadają się do badań wymagających pracy w czasie rzeczywistym, np. w pętach sprzętowych. Dodatkowo pozwalają na lepsze wyczucie dynamiki rzeczywistego obiektu niż w wypadku wersji referencyjnej symulatora, co również ma wysokie walory edukacyjne. Zaprezentowano także koncepcję symulatora sieciowego, który jest tematem dalszych prac autorów. Analizy funkcjonalno-sprzętowe wszystkich przedstawionych symulatorów względem kilku różnorodnych kryteriów, m.in. możliwości operatorskich, dostępności czy ceny, pozwoliły na rozdział ich zastosowań na typowo akademickie oraz popularyzatorskie.

Podziękowania

Praca powstała przy wsparciu z projektu Unii Europejskiej „INSPE – INNOVATIVE NUCLEAR AND SUSTAINABLE POWER ENGINEERING”. Autorzy pragną wyrazić swoje podziękowania za wsparcie.

Bibliografia

- Baum G. i in., Modele symulacyjne procesów zachodzących w podstawowych urządzeniach obiegu pierwotnego w warunkach normalnej eksploatacji i niedużych zakłóceń (SYMREP), Instytut Elektroenergetyki i Automatyki Politechniki Gdańskiej, 1989.
- Karla T., Tarnawski J., Duzinkiewicz K., Symulator czasu jądrowego procesów reaktora jądrowego [w:] Aktualne problemy automatyki i robotyki, red. Malinowski K., Józefczyk J., Świątka J., Akadematyczna Oficyna Wydawnicza EXIT, 2014, s. 558–569.
- Duzinkiewicz K., Cimiński A., Michalczyk Ł., Symulator szybkich procesów dynamiki reaktora jądrowego wodnego ciśnieniowego, *Pomiary · Automatyka · Robotyka PAR* 2013, nr 9, s. 97–101.
- Puchalski B. i in., Programowo-sprzętowa platforma symulacyjna – *Hardware In the Loop* – zaawansowanego układu sterowania poziomem wody w pionowej wylotnicy pary elektrowni jądrowej, [w:] Aktualne problemy automatyki i robotyki, red. Malinowski K., Józefczyk J., Świątka J., Akadematyczna Oficyna Wydawnicza EXIT, 2014, s. 570–580.
- Oka Y., Suzuki K., Nuclear Reactor Kinetics and Plant Control, Springer 2013.
- Hetrick D.L., Dynamics of Nuclear Reactors, The University of Chicago Press, Chicago and London, 1971.
- Sterling L., Taveter K., The Art of Agent-Oriented Modeling, The MIT Press, 2009.
- Seok H. i in., Development of software for the microsimulator for the KO-RI nuclear power plant unit 2, *Nuclear Technology* 1994, nr 106 (3), s. 384–396.
- Kluge A. i in., Designing training for process control simulators: a review of empirical findings and current practices, *Theoretical Issues in Ergonomics Science* 2009, nr 10 (6), s. 489–509.
- Thermal power plant simulation and control, Flynn D. (red.), *Power and Energy Series* 2003, No. 43.
- Raspberry PI Website [online], <http://www.raspberrypi.org> [dostęp: 30.10.2014].

Tomasz Karla

mgr inż.

Politechnika Gdańska | Politechnika Warszawska

e-mail: tomasz.karla@pg.gda.pl

Doktorant Politechniki Gdańskiej na Wydziale Elektrotechniki i Automatyki oraz Politechniki Warszawskiej na Wydziale Mechanicznym Energetyki i Lotnictwa w ramach projektu INSPE. Prowadzi badania z dziedziny sterowania wieloagentowego w zastosowaniu do obiegu pierwotnego elektrowni jądrowej z reaktorem typu PWR.

Jarosław Tarnawski

dr inż.

Politechnika Gdańska

e-mail: jaroslaw.tarnawski@pg.gda.pl

Absolwent Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej (2000), po studiach rozpoczął studia doktoranckie z dziedziny zastosowań sterowania adaptacyjnego do sterowania jakością w sieciach dystrybucji wody pitnej. Tytuł doktora otrzymał z wyróżnieniem (2006). Prowadzi prace badawcze w kierunkach implementacji zaawansowanych metod sterowania, decentralizacji sterowania, systemów wieloagentowych.

Kazimierz Duzinkiewicz

dr hab. inż.

Politechnika Gdańska

e-mail: kazimierz.duzinkiewicz@pg.gda.pl

Absolwent Wydziału Elektrycznego Politechniki Gdańskiej, specjalność automatyka i metrologia elektryczna. W 1983 roku obronił doktorat na macierzystej uczelni, a w 2009 roku uzyskał stopień doktora habilitowanego na Wydziale Elektrotechniki, Automatyki, Informatyki i Elektroniki Akademii Górniczo-Hutniczej w Krakowie. Obszar zainteresowań to: modelowanie i estymacja w warunkach niepewności, struktury i algorytmy sterowania, sterowanie optymalizujące wielkimi systemami. Autor monografii i skryptu dydaktycznego. Napisał ponad sto artykułów w recenzowanych czasopismach zagranicznych, krajowych, recenzowanych materiałach konferencji. Kierował jednym międzynarodowym projektem naukowo-badawczym, sześcioma krajowymi. Był głównym wykonawcą i wykonawcą w 18 projektach naukowo-badawczych lub badawczo-rozwojowych.