

## REALIZACJE SPRZĘTOWO-PROGRAMOWE SYMULATORÓW PODSTAWOWYCH ZASAD PRZEBIEGU PROCESÓW REAKTORA JĄDROWEGO

Tomasz KARLA<sup>1</sup>, Jarosław TARNAWSKI<sup>2</sup>, Kazimierz DUZINKIEWICZ<sup>3</sup>

Katedra Inżynierii Systemów Sterowania, Wydział Elektrotechniki i Automatyki, Politechnika Gdańska

1. tel.: (58) 347-17-42; e-mail: t.karla@eia.pg.gda.pl,

2. tel.: (58) 347-12-26; e-mail: jaroslaw.tarnawski@pg.gda.pl,

3. tel.: (58) 347-22-39; e-mail: kazimierz.duzinkiewicz@pg.gda.pl

**Streszczenie:** Przedstawiono możliwości i proces budowy symulatorów podstawowych zasad działania procesów reaktora jądrowego w oparciu o modele punktowe: kinetyki neutronów, procesów generacji i wymiany ciepła oraz procesów zatrutowania i wypalania paliwa. Referencyjny symulator został opracowany w środowisku MATLAB/Simulink. Na jego podstawie zbudowano w środowisku xPC drugi symulator pracujący w czasie rzeczywistym. Po dekompozycji funkcjonalnej reaktora jądrowego wydzielono procesy o różnych skalach czasowych i zaimplementowano wraz z mechanizmem synchronizacji w jednostkach Raspberry PI uzyskując rozproszoną wersję symulatora. W artykule rozpatrywany jest również sieciowy symulator obsługiwany z poziomu przeglądarki internetowej. Przeprowadzono weryfikację poprawności działania symulatorów. Dokonano porównania funkcjonalności i możliwych zastosowań symulatorów.

**Słowa kluczowe:** symulatory, reaktor jądrowy, kształcenie, praca w czasie rzeczywistym

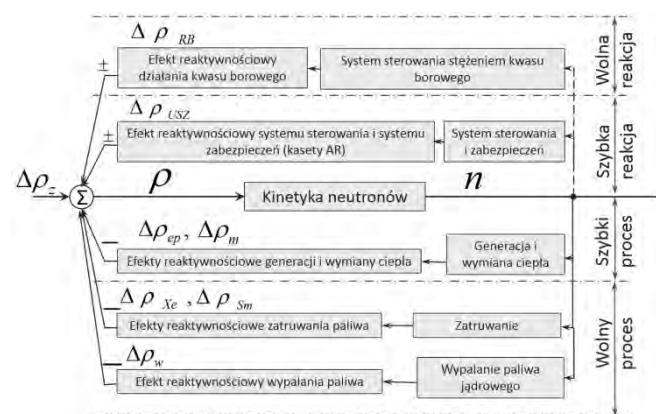
### 1. WPROWADZENIE

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 współzależnością. Tematykę symulatorów podejmowano już m.in. w [2, 3]. W artykule rozpatrywane są różne implementacje sprzętowo-programowe symulatorów podstawowych procesów reaktora jądrowego. W oparciu o modele punktowe procesów rozpatrywane są symulatory w oparciu o pakiet Matlab/Simulink, system czasu rzeczywistego xPC, komputery 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 elektrowni jest przedstawiany jako źródło ciepła o regulowanej wartości mocy [8]. Na pracę reaktora jądrowego ma wpływ szereg procesów. Do ich zamodelowania można użyć modeli o różnej złożoności opisanych np. w [5, 6]. Najprost-

szym 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 o modele punktowe [1]. Uwzględnione w symulatorach procesy i oddziaływania sterowania reaktorem zostały przedstawione na rys. 1.



Rys. 1. Główne procesy zachodzące w RJ wpływające na jego stan

Do pierwszej grupy zaliczają się reakcje jądrowe (kinetyka neutronów, zatrutowanie i wypalanie paliwa) i procesy wymiany ciepła. Kinetyka neutronów uwzględnia zmiany koncentracji neutronów ( $n$ ) w objętości rdzenia w zależności od reaktywności rdzenia ( $\rho$ ). Procesy zatrutowania i wypalania paliwa przedstawiają zmiany reaktywności ( $\Delta\rho_{(Xe,Sm)}$ ) reaktora związane z powstawaniem ksenonu ( $Xe-135$ ), jodu ( $J-135$ ), 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 z jakimi zachodzą. Tabela 1 przedstawia zestawienie poszczególnych procesów i ich czasów przejściowych. Różnice między najwolniejszymi i najszybszymi procesami sięgają kilku kilkunastu rzędów wielkości. Dlatego też dzieli się je dodatkowo na procesy szybkie i wolne (rys. 1).

Tablica 1. Przedziały czasowe poszczególnych procesów RJ

Proces	Czasy procesów przejściowych
Kinetyka neutronowa	10–5 do 10–3 s
Procesy wymiany ciepła między paliwem a chłodziwem/moderatorem, Zmiana reaktywności związana ze zmianami temperatury	3 do 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 godzin
Zatrucie samarem	15 do 20 dni
Wypalanie paliwa	dziesiątki dni / miesiące

### 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 o niskobudżetowe urządzenia Raspberry PI (RPI) [9] 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 o algorytm Bogacki-Shampine. Obliczenia były wykonywane w trybie freerun (bez stawiania wymagań czasowych dla obliczeń). Symulator udostępnia podgląd wszystkich wielkości procesowych w postaci dedykowanych bloków 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.

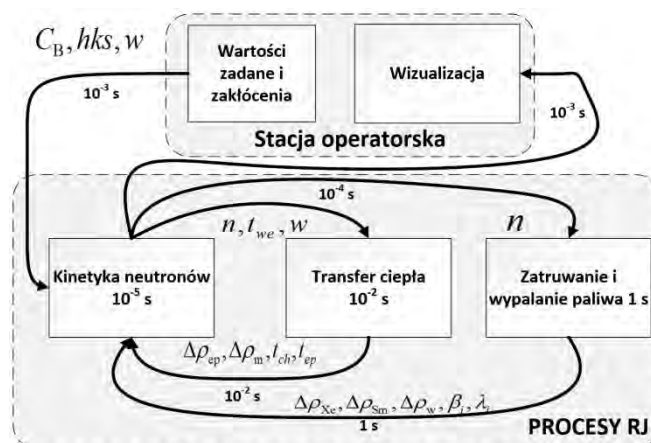
#### 3.2. Symulatory RT

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 odpowiadającemu średniemu czasowi życia neutronów) obliczenia dla danych chwil czasowych były wykonywane od około 10 razy wolniej do 100 razy szybciej

w stosunku to czasu rzeczywistego. Środowisko MATLAB/Simulink posiada przybory pozwalające na wykonywanie symulacji z zachowaniem pracy w czasie rzeczywistym. Autorzy rozpatrywali przybory Real-Time Windows Target (RTWT) i xPC, pozwalające na symulację na komputerach klasy PC oraz przybory zawierający pakiet modułów do współpracy i kompilowania aplikacji na platformę RPI.

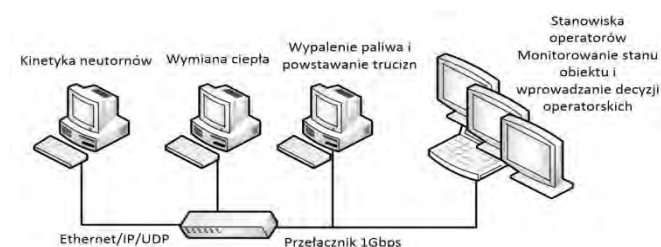
##### 3.2.1. Symulator RT o twardych ograniczeniach czasowych i małym kroku symulacji

Przybory xPC, zorientowany na symulację w trybie RT, oferuje dużą 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 3 komputery klasy PC. Dekompozycja czasowa procesów reaktora jądrowego przedstawiona w tabeli 1 pozwoliła na pogrupowanie poszczególnych procesów w bloki, które mogły być wykonywane z większymi krokami obliczeń (rys. 2).



Rys. 2. Symulator w wersji rozproszonej

Rozwiązanie takie wymagało jednak zapewnienia komunikacji między poszczególnymi urządzeniami oraz synchronizacji każdego z bloków symulacji z pozostałymi. Opracowano metodę synchronizacji momentów rozpoczynania, wstrzymywania i kończenia obliczeń na poszczególnych urządzeniach. Wszystkie jednostki były połączone ze sobą za pomocą gigabitowego przełącznika Ethernet przy wykorzystaniu bezpołączeniowego protokołu UDP (rys. 3).



Rys. 3. Realizacja sprzętowa symulatora w wersji zdecentralizowanej

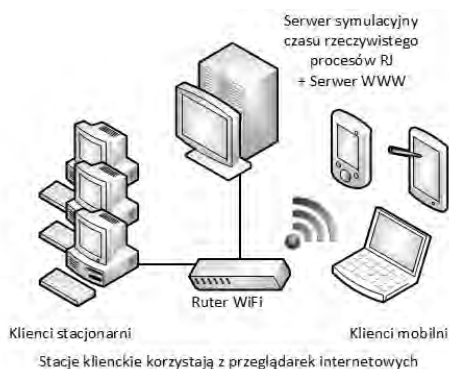
##### 3.2.2. Symulator RT o miękkich ograniczeniach czasowych w RPI

Za pomocą przybory Simulink Support Package for Raspberry PI wygenerowano aplikację symulatora RT dla RPI typu *stand-alone* uruchamianą bez użycia M/S. Jako docelowy krok obliczeń przyjęto 1E-3s (górny próg ze śred-

niego czasu życia neutronów). 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. Wdrożono mechanizmy synchronizacji symulacji na poszczególnych urządzeniach. Opracowano także w języku C niezależną od M/S aplikację wizualizacyjną, która mogła być uruchamiana na czwartym RPI. Do komunikacji między wszystkimi urządzeniami użyto połączenia przez interfejs Ethernet z wykorzystaniem gigabitowego przełącznika i protokołu UDP.

### 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. 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 PCta, laptopa, tabletu, czy nawet telefonu komórkowego. Idea tego rozwiązania jest zaprezentowana na rys. 4.



Rys. 4. Symulator w wersji sieciowej

### 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. Drugi etap weryfikacji polegał na porównaniu wyników symulatora referencyjnego oraz RT w oparciu o te same, wcześniej przygotowane scenariusze przebiegów zmiennych sterujących i zakłóceń. Według wzoru (1) wyznaczono średni błąd symulacji dla każdego z symulatorów RT. Wyniki procesu weryfikacji przedstawia tabela 2.

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

gdzie:  $w_r(i)$  – wartość referencyjna z symulatora nie pracującego w RT,

$w_s(i)$  – pomiar z symulatora RT

$N$  – liczba pomiarów.

Tabela 2. Zestawienie wyników procesu weryfikacji symulatorów

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

## 4. ZESTAWIENIE CECH SYMULATORÓW

Przyjęto kilka kryteriów jakościowych, pod względem których oceniano poszczególne symulatory. Zestawienia cech symulatorów dokonano w tabelicy 3. Kryterium Zgodność, określana jest na podstawie wyników symulacji opisanych w rozdziale 3.3. Kolejnym kryterium jest możliwość pracy w trybie RT, co znacznie podwyższa poczucie pracy z rzeczywistym obiektem oddając jego rzeczywistą dynamikę.

Tabela 3. Zestawienie cech przedstawionych symulatorów

	Zgodność	Real-Time	Opcje operatorskie	Archiwizacja	Scenariusze	HMI	HIL	Rozposzcz-nienie	Koszt
Matlab/Simulink	wzór	brak	oparte o kontroli dostępne w M/S	wysoki	wysoki	średni	brak	średni	średni
xPC	wysoka	twardy	proste, ograniczone trudności	średni	średni	średni	średni	niski	wysoki
RPI	średnia	twardy/miękki	autorska aplikacja dla RPI	średni	średni	wysoki	wysoki	bardzo wysoki	średni/niski
Steciowy	średnia	miękki	elastyczne, oparte o kontroli przeglądarki	wysoki	średni	wysoki	brak	wysoki	niski

Kryterium określające rozbudowanie możliwości wpływania na parametry i zmienne obiektu określono jako Opcje operatorskie. 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 (HIL – *Hardware in the Loop*) opisanych w [4]. Dostępność rozwiązań, urządzeń i oprogramowania oraz popularność narzędzi zawarto pod kryterium Rozposzcz-nienie. Ostatnim wziętym pod uwagę kryterium jest Koszt implementacji symulatora.

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. Największe możliwości kształtowania scenariuszy pracy posiada symulator referencyjny M/S. 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. WNIOSKI KOŃCOWE

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 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. Symulatory RT pozwalają na lepsze wyczuć dynamiki rzeczywistego obiektu niż w wypadku wersji referencyjnej symulatora co posiada wysokie walory edukacyjne. Zaprezentowano także koncepcję symulatora sieciowego, który jest tematem dalszych prac autorów. Analizy wszystkich przedstawionych symulatorów względem kilku różnorodnych kryteriów m. in. możliwości operatorskich, dostępności czy ceny pozwoliły na rozdział ich zastosowań na typowo akademickie oraz popularyzatorskie.

## 6. BIBLIOGRAFIA

1. Baum G., Duzinkiewicz K., Michalak A., Wojtoń K., 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. Karla T., Tarnawski J., Duzinkiewicz K., Symulator czasu rzeczywistego procesów reaktora jądrowego, Aktualne Problemy Automatyki i Robotyki, Akademicka Oficyna Wydawnicza EXIT, 2014, s. 558–569
3. Duzinkiewicz K., Cimiński A., Michalczyk Ł., Symulator szybkich procesów dynamiki reaktora jądrowego wodnego ciśnieniowego, Pomiary Automatyka Robotyka, 9/2013, s. 97–101.
4. Puchalski B., Rutkowski T., Tarnawski J., Duzinkiewicz K., Programowo-sprzętowa platforma symulacyjna – Hardware In the Loop – zaawansowanego układu sterowania poziomem wody w pionowej wytwornicy pary elektrowni jądrowej, Aktualne Problemy Automatyki i Robotyki, Akademicka Oficyna Wydawnicza EXIT, 2014, s. 570–580
5. Oka Y., Suzuki K., Nuclear Reactor Kinetics and Plant Control, Springer, 2013
6. Hetrick D. L., Dynamics of Nuclear Reactors, The University of Chicago Press, Chicago and London, 1971
7. Sterling L. Taveter K., The Art of Agent-Oriented Modeling, The MIT Press, 2009
8. Flynn, D., Thermal power plant simulation and control (No. 43). IET, 2003.
9. Raspberry PI Website <http://www.raspberrypi.org>, Dostęp 30.10.2014

## HARDWARE-SOFTWARE IMPLEMENTATION OF BASIC PRINCIPLES SIMULATOR OF NUCLEAR REACTOR PROCESSES

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.

**Keywords:** modeling, simulation, nuclear reactor, real-time requirements, education, training