

## ADAPTACYJNY DOBÓR KROKU OBLICZEŃ DLA SYMULATORA CZASU RZECZYWISTEGO O MIĘKKICH WYMAGANIACH CZASOWYCH

Jarosław TARNAWSKI<sup>1</sup>, Tomasz KARLA<sup>2</sup>, Tomasz A. RUTKOWSKI<sup>3</sup>,  
Bartosz PUCHALSKI<sup>4</sup>, Kazimierz DUZINKIEWICZ<sup>5</sup>

Politechnika Gdańska, Wydział Elektrotechniki i Automatyki

1. tel.: 58 347-12-26 e-mail: jaroslaw.tarnawski@pg.gda.pl  
2. tel.: 58 347-17-42 e-mail: tomasz.karla@pg.gda.pl  
3. tel.: 58 347-12-26 e-mail: tomasz.adam.rutkowski@pg.gda.pl  
4. tel.: 58 347-17-42 e-mail: bartosz.puchalski@pg.gda.pl  
5. tel.: 58 347-22-39 e-mail: kazimierz.duzinkiewicz@pg.gda.pl

**Streszczenie:** Symulacje czasu rzeczywistego mają duże walory praktyczne i dydaktyczne w zastosowaniach automatyki. Praca z symulatorem czasu rzeczywistego pozwala na poznanie i doświadczenie dynamiki symulowanego obiektu, wprowadzanie sterowań oraz decyzji operatorskich w warunkach czasowych odpowiadających rzeczywistemu obiektowi. Realizacja symulatorów czasu rzeczywistego wymaga zakończenia obliczeń numerycznych jednego kroku symulatora w czasie krótszym niż założona długość kroku symulacji. Jest to wyzwanie zwłaszcza dla małych kroków symulacji. Innego typu problemem jest realizacja symulatora czasu rzeczywistego o wymaganiach miękkich (tj. z możliwymi incydentalnymi naruszeniami ograniczeń na czas obliczeń numerycznych) w środowisku obliczeniowym nie będącym systemem czasu rzeczywistego. Wobec powyższych wymagań i ograniczeń konieczne jest rozwiązanie zawierające kompromis pomiędzy nadążaniem za czasem symulacji kosztem wydłużania kroku symulacji co może powodować pogorszenie wyników symulacji. W artykule rozpatrywane jest zagadnienie adaptacyjnego doboru kroku symulacji zastosowane dla symulatora procesów reaktora jądrowego.

**Słowa kluczowe:** symulacja czasu rzeczywistego, adaptacyjny dobór kroku symulacji, symulacja procesów reaktora jądrowego.

### 1. WSTĘP

Symulatory czasu rzeczywistego opierają się na modelach matematycznych opracowanych w celu odwzorowania zachowania symulowanego obiektu w maszynie cyfrowej. Mogą być one wykorzystane do symulacji w tzw. pętli sprzętowej i zagadnieniach szybkiego prototypowania [3]. W większości modeli matematycznych wykorzystuje się układy równań różniczkowych zwyczajnych, które są rozwiązywane przez odpowiednie metody numeryczne. Metody te można podzielić na: metody ze stałym i zmiennym krokiem. Stały krok oznacza, że dla całego zakładanego horyzontu czasowego symulacji system równań różniczkowych jest rozwiązywany dla stałych odstępów czasu, natomiast zmienny krok oznacza, że rozwiązania nie muszą być oddalone od siebie o jednakową jednostkę czasu. W przypadku rozwiązywania układów równań różniczkowych, gdy nie jest wymagana praca w warunkach czasu rzeczywistego, w większości przypadków

wybiera się metody numeryczne ze zmiennym krokiem. Są one stabilniejsze, mają większą precyzję, są bardziej popularne oraz często występują jako domyślne w środowiskach szybkiego prototypowania takich jak MATLAB/Simulink [4, 5]. Nieocenioną zaletą metod numerycznych ze zmiennym krokiem jest możliwość prawidłowego rozwiązania układu równań sztywnych. Systemy opisane równaniami sztywnymi posiadają w znacznym stopniu różniące się stałe czasowe (szybka i wolna dynamika). Stałe czasowe reprezentujące szybką dynamikę mogą być pobudzone przez metodę rozwiązywania równań prowadząc do niestabilności numerycznej całego systemu. Problemem są również zbyt długie czasy trwania obliczeń w przypadku wykorzystania metod ze stałym krokiem przy stosowaniu małych interwałów czasowych [4]. W przypadku, gdy wymagana jest praca systemu w czasie rzeczywistym najczęściej wybierane są metody ze stałym krokiem, gdyż są one łatwiejsze w implementacji oraz mają jednakowy czas obliczeń pomiędzy dyskretnymi chwilami czasu [4].

Obecne prace związane z zastosowaniem metod numerycznych ze zmiennym krokiem w systemach pracujących w czasie rzeczywistym nie są licznie reprezentowane w literaturze. W artykule [6] autorzy poruszają kwestię metod numerycznych ze zmiennym krokiem w kontekście czasu rzeczywistego w zastosowaniu do symulatora lotu. Jednakże w tej publikacji nie jest wykorzystywany mechanizm adaptacji kroku obliczeń.

Motywacją dla podjęcia tematu rozpatrywanego w artykule była chęć zbadania wpływu metody numerycznej ze zmiennym, adaptowanym krokiem na jakość obliczeń oraz przydatność tej metody w przypadku obliczeń wykonywanych w warunkach czasu rzeczywistego o miękkich wymaganiach czasowych w środowisku obliczeniowym nie będącym systemem czasu rzeczywistego, np. symulator, w którym interfejs użytkownika będzie zrealizowany w przeglądarce internetowej.

Zaimplementowany w serwerze kod symulatora może być wykorzystany do prezentacji wielu użytkownikom jednocześnie (z wykorzystaniem wielu typów urządzeń – w tym mobilnych) tego samego zagadnienia i wczytywania

decyzji operatorskich od uprawnionych użytkowników. Innym sposobem wykorzystania jest praca na wielu instancjach symulatora, czyli niezależne wykorzystanie kodu symulatora dla każdego z użytkowników. Oba te zastosowania mają bardzo duży potencjał dydaktyczny i popularyzatorski. Jednak ich immanentną cechą realizacji w strukturze serwer-klient (np. przeglądarka internetowa) będzie zróżnicowany czas przeznaczony/dostępny na obliczenia w danym kroku symulacji oraz dostarczenie wyników do przeglądarki. Wobec braku gwarancji zrealizowania i dostarczenia wyników w założonym *a priori* czasie spodziewane są przekroczenia kroku symulacji i tym samym potrzeba "dogonienia" czasu rzeczywistego. Jak już wcześniej zaznaczono rozpatrywanym w tym artykule rozwiązaniem tego problemu jest stosowanie zmiennego kroku symulacji i adaptacyjny dobór jego długości.

## 2. ADAPTACYJNY DOBÓR KROKU SYMULACJI

### 2.1. Wyznaczanie czasu trwania obliczeń jednego kroku

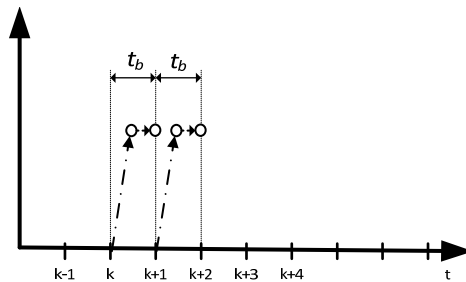
Oszacowanie czasu trwania jednego kroku symulacji może być zrealizowane w różny sposób. Najprostszą metodą jest wyznaczenie czasu trwania obliczeń w poprzednim kroku symulacji i założenie, że w kroku następnym obliczenia będą trwały tyle samo. Innym prostym sposobem jest uruchomienie procedury obliczeniowej, wyznaczenie czasu potrzebnego na wykonanie całej symulacji i uwzględniając liczbę kroków można wyznaczyć średni czas kroku. Pierwsze rozwiązanie pomimo, że prostsze wydaje się korzystniejsze, gdyż typowym problemem w systemach nie pracujących w warunkach czasu rzeczywistego jest występujące co pewien czas „przytkanie” spowodowane przydzieleniem zasobów obliczeniowych innym ważniejszym z punktu widzenia systemu operacyjnego zadaniom. Metoda druga uśredniałaby rzadko występujące przekroczenia oszacowując długość każdego kroku bardziej konserwatywnie.

### 2.2. Nadążanie za czasem rzeczywistym i dobór kroku symulacji

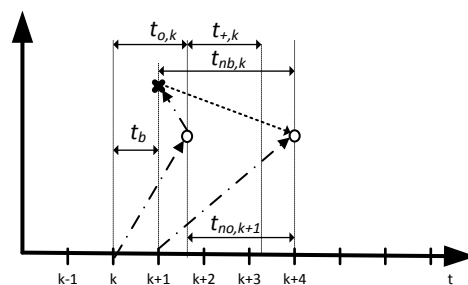
Sytuację, w której obliczenia zakończyły się przed upływem czasu kroku bazowego przedstawiono na rysunku 1. Jednak w przypadku symulacji *real-time* w środowisku nie spełniającym twardych wymagań czasu rzeczywistego jest wysoce prawdopodobne przekraczanie czasu obliczeń dla jednego kroku symulacji. Należy wówczas podjąć decyzję co do strategii postępowania. Strategia pierwsza zakłada kontynuowanie obliczeń ze stałym krokiem, co wiąże się z opóźnianym generowaniem wyników, kumulowanym w kolejnych krokach. To rozwiązanie nazwano w artykule *Soft Real Time* – SRT. Alternatywą jest metoda ze zmiennym adaptacyjnie dobieranym krokiem symulacji. Jej zaletą jest „nadążanie” za czasem rzeczywistym symulacji kosztem wydłużenia kroku symulacji i ewentualnego pogorszenia jakości obliczeń. To rozwiązanie nazwane jest *Adaptive Step Real Time* – ASRT.

Idea doboru kroku w podejściu ASRT została zaprezentowana na rysunku 2. Gdy obliczenia jednego kroku symulacji zakończą się przed upływem bazowego kroku, długość kroku symulacji nie jest zmieniana. Gdy jednak obliczenia będą trwały dłużej, kolejny krok symulacji zostanie wydłużony. Bazując na długości czasu obliczeń z obecnego kroku można obliczyć wartość symulowanej wielkości na taką chwilę w przyszłości, aby obliczenia

mogły być zakończone do tego czasu i pokrywały się z krokiem bazowym. Skutkuje to podawaniem symulowanych wyników zgodnie z wymaganiami czasu rzeczywistego jednak nie w każdej wielokrotności bazowego kroku. Gdy w kolejnych krokach obliczenia trwają krócej przywracany jest bazowy, wyjściowy rozmiar kroku.



Rys. 1. Zakończenie obliczeń przed upływem czasu kroku bazowego ( $t_b$  - czas obliczeń bazowych)



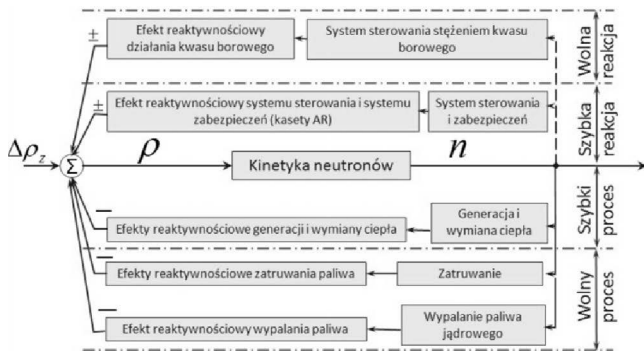
Rys. 2. Metoda doganiania czasu rzeczywistego podczas symulacji ( $t_b$  - czas obliczeń bazowych,  $t_{o,k}$  - czas obliczeń w kroku  $k$ , który przekracza czas obliczeń bazowych,  $t_{+,k}$  - szacowany czas potrzebny na wykonanie obliczeń,  $t_{nb,k}$  - adaptacyjnie dobrany krok obliczeń,  $t_{no,k+1}$  - nowy czas obliczeń umożliwiający synchronizację w przyszłości z krokiem obliczeń bazowych)

## 3. SYMULATOR PROCESÓW REAKTORA JĄDROWEGO Z ADAPTOWANYM KROKIEM

Zaprezentowany w rozdziale 2 mechanizm adaptacyjnego doboru długości kroku symulacji został zaimplementowany w symulatorze czasu rzeczywistego procesów reaktora jądrowego, którego podstawy teoretyczne (opis modelu) i budowę przedstawiono w [1]. Jest to rozbudowany symulator złożony z 18 równań różniczkowych oraz 32 równań arytmetycznych. Zjawisko kinetyki neutronów powinno być odwzorowane z krokiem co najmniej  $10^{-5}$  s. Niezakłócona symulacja z takim krokiem to wyzwanie nawet dla współczesnych maszyn cyfrowych i autorzy wielokrotnie odnotowali naruszenia kroku symulacji w implementacjach symulatora na różnych platformach sprzętowo-programowych [2]. Potrzeba zastosowania mechanizmu doboru kroku jest w tym symulatorze wyraźnie potwierdzona doświadczeniami.

### 3.1. Charakterystyka symulowanych procesów

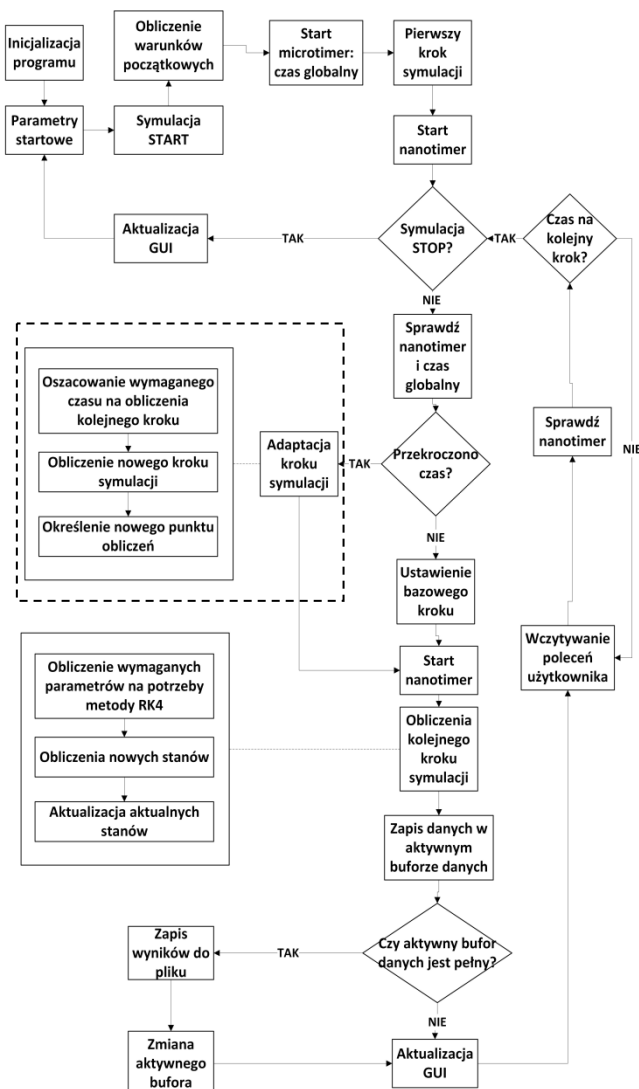
W symulatorze rozpatrywane są procesy kinetyki neutronów, generacji i wymiany ciepła, zatrufania i wypalania paliwa oraz efekty reaktywnościowe związane ze sterowaniem reaktorem. Wymienione procesy symulatora przedstawiono na rysunku 3.



Rys. 3. Procesy reaktora jądrowego odwzorowane w symulatorze ( $\rho$  – reaktywność reaktora,  $\Delta\rho_z$  – reaktywność zewnętrzna,  $n$  – gęstość neutronów)

### 3.2. Realizacja programowa symulatora

Implementacja symulatora czasu rzeczywistego zawierająca mechanizm adaptacyjnego doboru długości kroku pracy zrealizowana została wg algorytmu przedstawionego na rysunku 4.



Rys. 4. Algorytm symulatora czasu rzeczywistego z metodą adaptacyjnego doboru długości kroku

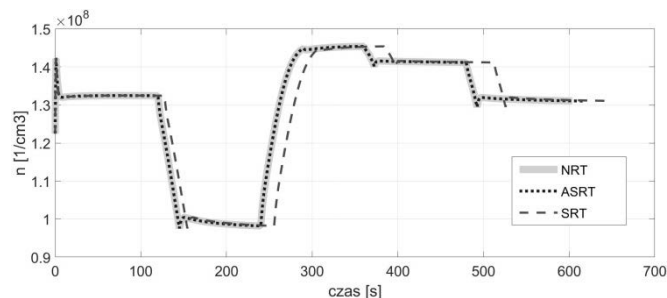
W ramce oznaczonej linią przerywaną znajduje się najistotniejsza część metody adaptacyjnego doboru kroku symulacji.

### 3.3. Ocena jakości działania symulatora z adaptacyjnym doborem długości kroku

Na potrzeby weryfikacji opracowanych wersji symulatora przygotowano symulator referencyjny zbudowany w środowisku MATLAB/Simulink i nazwano go *Non Real Time* - NRT. Scenariusz pracy reaktora użyty do testów zakładał cykliczną zmianę położenia prętów sterujących co 120 sekund wg wzorca: 175 cm, 125 cm, 225 cm, 200 cm, 175 cm. Warunki początkowe zakładały pracę z mocą cieplną równą 100% mocy nominalnej, przy wszystkich innych parametrach nominalnych dostosowanych dla tej mocy. Czas trwania symulacji ustawiono na 10 minut. Symulator referencyjny oparto o metodę Rungego-Kutty czwartego rzędu ze zmiennym krokiem. Na podstawie otrzymanych danych można oceniać rozbieżności wyników różnych implementacji (SRT i ASRT) wynikających głównie z długości kroku i punktualności generowania danych.

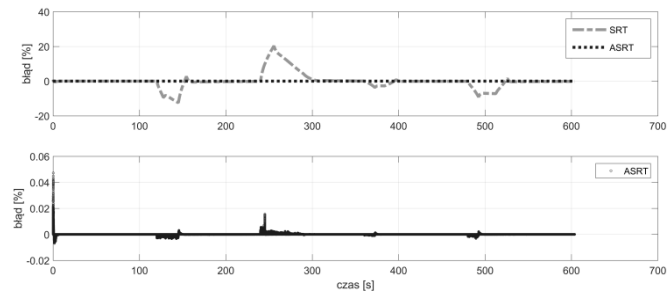
Symulatory SRT i ASRT zostały zbudowane w języku C++ z wykorzystaniem środowiska Visual Studio 2015. Symulacje wykonywano w środowisku Windows 10 x64, na komputerze PC z procesorem klasy i7. Bazowy krok symulacji wynosił  $1e-5$  s.

Na rysunku 5 przedstawiono wyniki symulacji wybranej wielkości dla symulatorów NRT, SRT i ASRT. W tej skali przebiegi NRT i ASRT nakładają się. Wyraźnie widać przesunięcie w czasie wyników symulatora SRT.



Rys. 5. Przebiegi gęstości neutronów  $[n/cm^3]$  w symulowanym reaktorze jądrowym dla symulatorów NRT, SRT i ASRT

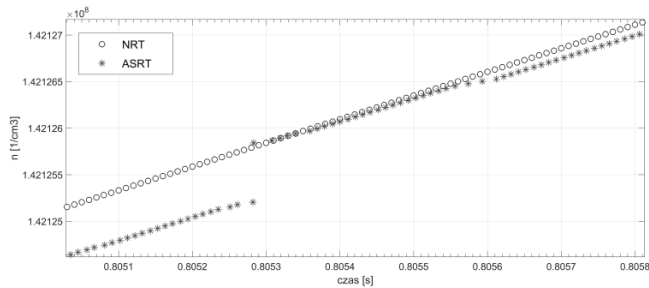
Na rysunku 6 zaprezentowano błędy symulatorów SRT i ASRT względem NRT. Można zauważyć, iż błąd generowany przez symulator SRT jest istotnie większy od symulatora ASRT. Maksymalny błąd symulatora SRT wynosi około 25%, a błąd symulatora ASRT nie przekraczał 0,05% na początku symulacji, a następnie był jeszcze mniejszy.



Rys. 6. Wykresy błędów względnego symulatorów SRT i ASRT odniesione do symulatora referencyjnego NRT (górny wykres) i powiększony wykres błędów symulatora ASRT (dolny wykres)

Na rysunku 7 przedstawiono krótki wycinek czasowy symulacji NRT i ASRT. Można zauważyć specyficzne

efekty działania symulatora ASRT. Na skutek wydłużania kroku symulacji nie otrzymano wyników w każdym kroku symulacji względem symulatora referencyjnego. Wydłużony krok pracy symulatora powoduje powstawanie błędów w stosunku do dokładnego rozwiązania NRT.



Rys. 7. Ilustracja mechanizmu działania adaptacyjnego doboru kroku w symulatorze ASRT względem NRT

Na podstawie zarejestrowanych wyników z symulatora referencyjnego oraz symulatorów SRT i ASRT obliczono błąd względny  $e_w$  wg zależności (1) a wyniki przedstawiono w tablicy 1.

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

gdzie:

$w_r(i)$  – wartości z symulatora referencyjnego NRT,  $w_s(i)$  – wartości z symulatorów SRT lub ASRT,  $N$  – liczba próbek

Tablica 1. Błędy względne symulatorów SRT i ASRT względem symulatora referencyjnego NRT

Wersja symulatora	Błąd względny $e_w$
SRT	2.0342 %
ASRT	71.9836e-006 %

#### 4. PODSUMOWANIE

Praca z symulatorem czasu rzeczywistego pozwala na poznanie dynamiki symulowanego obiektu, wprowadzanie sterowań oraz decyzji operatorskich w warunkach czasowych odpowiadających rzeczywistemu obiektowi. Problemem jest realizacja symulatora czasu rzeczywistego o wymaganiach miękkich (tj. z możliwymi incydentalnymi naruszeniami warunków czasu rzeczywistego) w środowisku obliczeniowym nie będącym systemem czasu rzeczywistego. Wobec wymagań zakończenia obliczeń jednego kroku symulatora w czasie krótszym niż długość kroku symulacji, konieczne jest rozwiązanie zawierające kompromis

między nadążaniem za czasem symulacji kosztem wydłużania kroku symulacji. W artykule rozpatrywane jest zagadnienie adaptacyjnego doboru długości kroku symulacji zastosowane dla symulatora procesów reaktora jądrowego. Wybrany obiekt testowy jest bardzo wymagający pod względem obliczeniowym. Opisana jest metoda kompensacji opóźnień obliczeniowych w celu utrzymania pracy w czasie rzeczywistym. Przeprowadzone badania wskazują wyższość metody z adaptacyjnym doбором długości kroku w porównaniu z symulatorem ze stałym krokiem. Istotnie mniejszy błąd względny w porównaniu do symulatora referencyjnego wskazuje na skuteczność proponowanej metody i poprawność jej implementacji.

#### 5. BIBLIOGRAFIA

1. 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), Miedzyzdroje 24-27 Aug. 2015, Page(s) 1074-1079. IEEE.
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. 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.
4. Popovici, K., & Mosterman, P. J. (Eds.). (2012). Real-time simulation technologies: principles, methodologies, and applications. CRC Press
5. Baracos, P., Murere, G., Rabbath, C. A., & Jin, W. (2001). Enabling PC-based HIL simulation for automotive applications. In Electric Machines and Drives Conference, 2001. IEMDC 2001. IEEE International (pp. 721-729). IEEE.
6. Lei, Z., & Hongzhou, J. (2012, December). Variable step euler method for real-time simulation. In Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on (pp. 2006-2010). IEEE.

### SOFT REAL-TIME SIMULATION WITH ADAPTIVE STEP OF COMPUTATION

Real time simulation is a very useful tool in the workshop of modern engineer because of its didactical and practical benefits. Working in real time with simulation models can lead to better understanding of the process through examination of its dynamics. Another very important benefit of real time simulation is a fact that operators decision are taken into account in the same time scale as the real system would work. Sometimes the hard real time regime is not obtainable in specified computational environment, for instance in client server architecture due to time delays and large amount of queries. In this case soft real time approach can be applied. In the paper authors present and validate a novel approach of soft real time solver that is able to adapt its time step to actual computational effort of the system. Approach presented in the paper is based on the idea of following after the real time when the computed time step is larger than a priori known global time step of the simulation. Presented work proves that this adaptive time step can improve system response compared to typical soft real time regime solver. This approach is validated with nuclear reactor model that requires significant computational effort because it has extremely fast dynamics.

**Keywords:** real-time simulation, adaptive step of computation, nuclear reactor processes simulator.