

SYMULACYJNA WERYFIKACJA ALGORYTMÓW W ŚRODOWISKU PTOLEMY II

Władysław SZCZEŚNIAK

Wyższa Szkoła Zarządzania w Gdańsku, 80-952 Gdańsk, ul Pelplińska 3

tel: 58 769 0 800, fax: 58 769 0 819, e-mail: wlad77@wp.pl

Streszczenie: Praca prezentuje domeny systemu PTOLEMY II, a bardziej szczegółowo przedstawiono w niej wybrane możliwości domeny Synchronous Data Flow (SDF) do weryfikacji algorytmów metodą symulacyjną. Pozytywny wynik symulacji decyduje o implementacji symulowanego algorytmu na przykład w systemach wbudowanych. Zaprezentowane rozwiązanie pozwala na szybsze uzyskanie prototypu, a także prowadzi do obniżenia kosztów projektowania.

Słowa kluczowe: systemy wbudowane, symulacja algorytmów, system Ptolemy

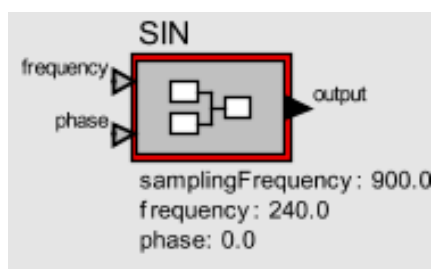
1. WPROWADZENIE

System PTOLEMY II został opracowany w University of California, Berkeley i jest przeznaczony dla platformy Windows. Jego poprzednikiem był system Ptolemy Classic pracujący na platformach Unixowo/Linuxowych. Oba systemy są public domain, co ma istotne znaczenie przy ich wykorzystaniu nie tylko do celów edukacyjnych.

2. WPROWADZENIE DO SYSTEMU PTOLEMY II

2.1. Instrukcja użytkownika systemu PTOLEMY II

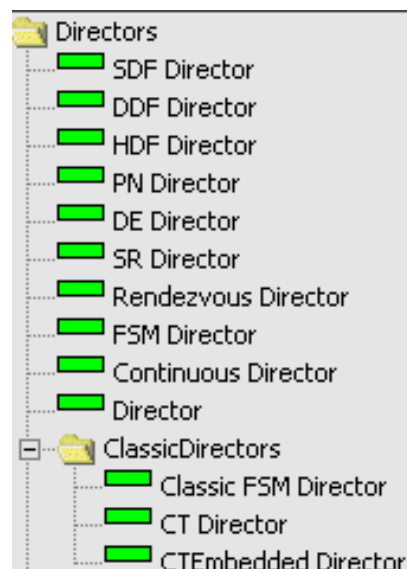
Trzytomowa instrukcja dotycząca systemu PTOLEMY II jest bardzo szczegółowa i liczy kilkaset stron. Natomiast dostęp do opisu wybranego modelu/aktora jest szybki i wygodny. Jest on uzyskiwany po prawym kliknięciu na interesujący użytkownika model. Przykład przedstawiono na rysunku 1.



Rys. 1. Przykładowy fragment opisu modelu SIN.

2.2. Wybór domeny w systemie PTOLEMY II

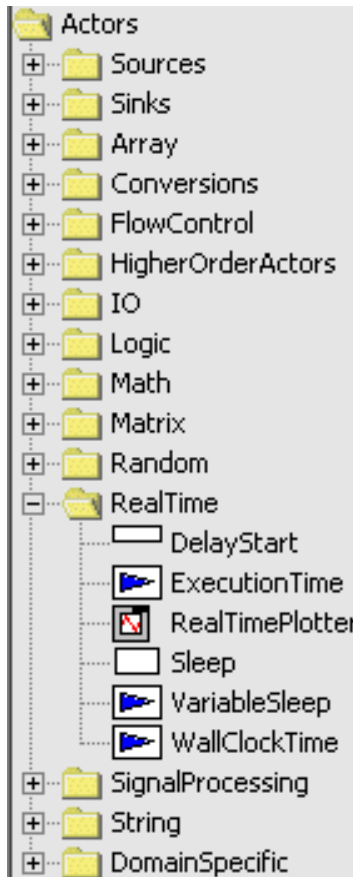
Po ściągnięciu systemu ze strony UC Berkeley [1], jego zainstalowaniu w środowisku Windows, a następnie uruchomieniu ptolemowego edytora Vergil i otwarciu katalogu **Directories** oraz podkatalogu **Classic Directories** zostaje uwidoczniona ich zawartość przedstawiona na rysunku 2. Autorzy systemu PTOLEMY II istniejące domeny przypisali nazwom Directories. Wybierając określoną domenę do symulacji algorytmu/systemu należy podczas edycji projektu umieścić na nim odpowiednią ikonę (np. SDF Director). Proces ten jednoznacznie określa domenę projektową ze zbioru zaprezentowanego na rysunku 2.



Rys. 2. Wykaz domeny systemu PTOLEMY II.

2.3. Edycja projektu

Przeznaczony do symulacji system/algorytm należy wyedytować przy pomocy Vergila z wykorzystaniem zbioru dostępnych modeli (ang. Actors), które znajdują się w katalogu **Actors** (rys. 3.). Po jego otwarciu dostępne są podkatalogi tematyczne z określonymi modelami stosowanymi w procesie symulacji.

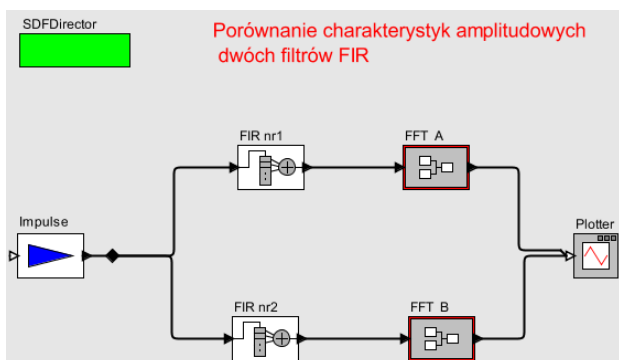


Rys. 3. Wykaz podkatalogów z modelami z wyszczególnieniem zawartości podkatalogu Real Time.

3. SYMULACJA WYBRANYCH ALGORYTMÓW W ŚRODOWISKU PTOLEMY II w DOMENIE SDF

3.1. Porównywanie algorytmów filtracji filtrów cyfrowych FIR

Przy pomocy domeny SDF systemu PTOLEMY II można bardzo szybko porównać charakterystyki projektowanych filtrów cyfrowych. Liczba porównywanych jednocześnie filtrów jest ograniczona głównie tylko czytelnością uzyskiwanych charakterystyk.

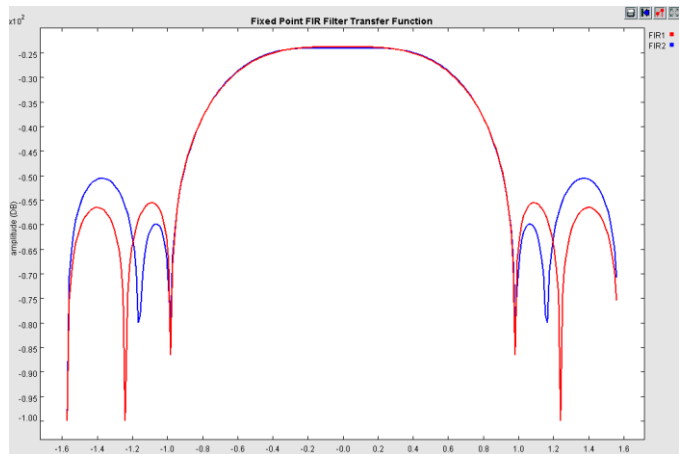


Rys. 4. Struktura symulowanego modelu z dwoma filrami FIR.

Rysunek 4. przedstawia prosty projekt umożliwiający porównanie dwóch algorytmów filtracji cyfrowej (FIR nr1 i FIR nr2) różniących się wartościami współczynników (w zasadzie występuje tutaj różnica w dokładności tychże współczynników).

Uruchomienie symulacji dla powyższego projektu daje charakterystyki amplitudowe uprzednio zaprojektowanych

filtrów FIR nr1 i FIR nr2, które są przedstawione na rys 5.

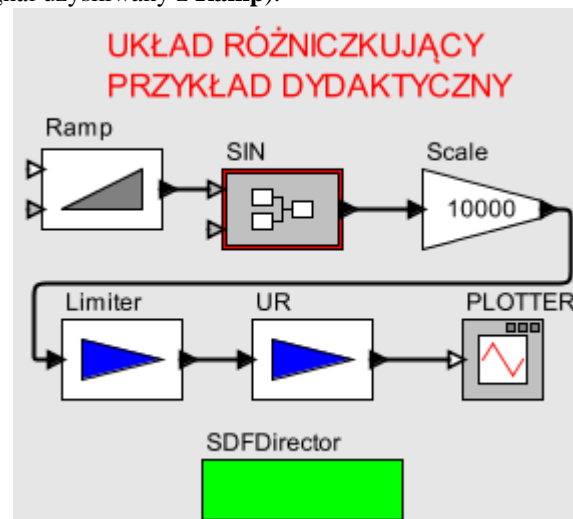


Rys. 5. Wynik symulacji projektu z rys. 4.

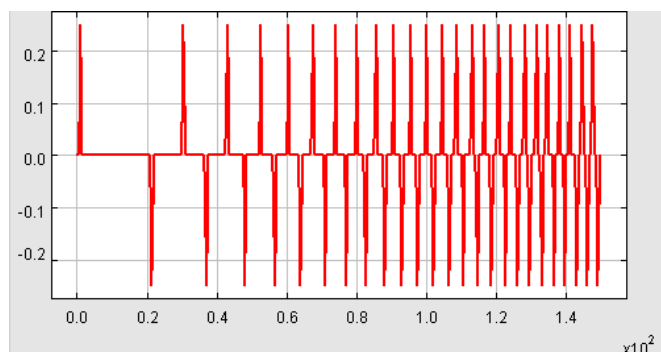
Na podstawie analizy uzyskanych charakterystyk można podjąć decyzję o akceptacji określonej wersji algorytmu do dalszego procesu projektowego, na przykład jego implementacji w systemie wbudowanym.

3.2. Weryfikacja procesu różniczkowania sygnałów

Poprawny dobór parametrów układu różniczkującego przy określonych częstotliwościach sygnału wejściowego decyduje o właściwościach projektowanego podsystemu. Na rysunku 6. zaprezentowano prosty układ symulujący działanie układu różniczkującego (UR) przy różnych częstotliwościach sygnału wejściowego (zmiana częstotliwości generatora sinusoidalnego -- SIN poprzez sygnał uzyskiwany z Ramp).

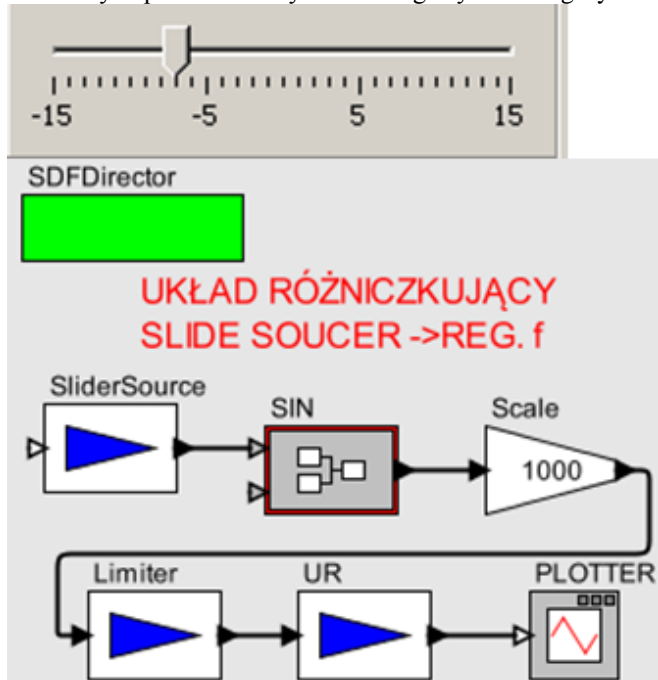


Rys. 6. Przykład struktury modelu układu różniczkującego.

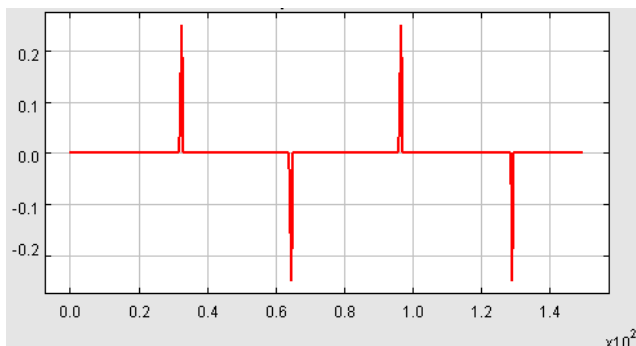


Rys. 7. Uzyskane wyniki symulacyjne układu różniczkującego.

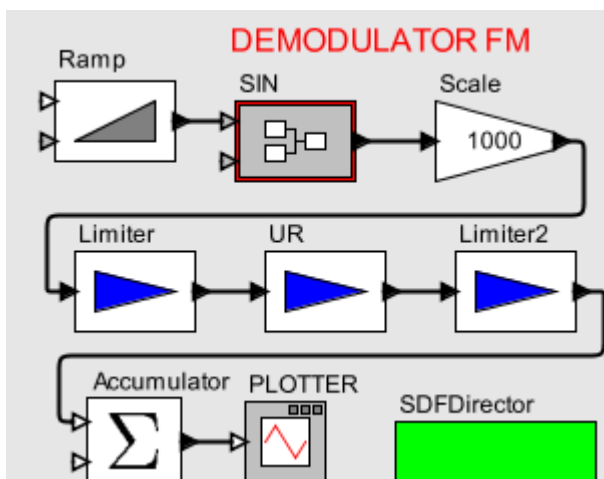
Struktura symulowanego układu zaprezentowana na rysunku 6. ma charakter dydaktyczny. Można ją zmodyfikować poprzez wprowadzenie ręcznego sterowania np. częstotliwością generatora SIN, przy pomocy aktora **SliderSource**, co widnieje na rysunku 8. Rozwiązanie to jest wygodne w przypadku konieczności uziemienniania określonych parametrów symulowanego systemu/algorytmu.



Rys. 8. Przykład struktury modelu układu różniczkującego z ręczną regulacją/ustawianiem częstotliwości generatora SIN przy pomocy SliderSoucer'a.

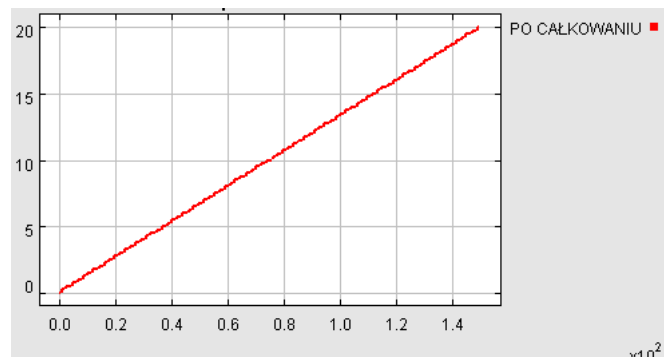


Rys. 9. Wyniki symulacyjne układu różniczkującego z rysunku 8.



Rys. 10. Przykład demodulatora FM.

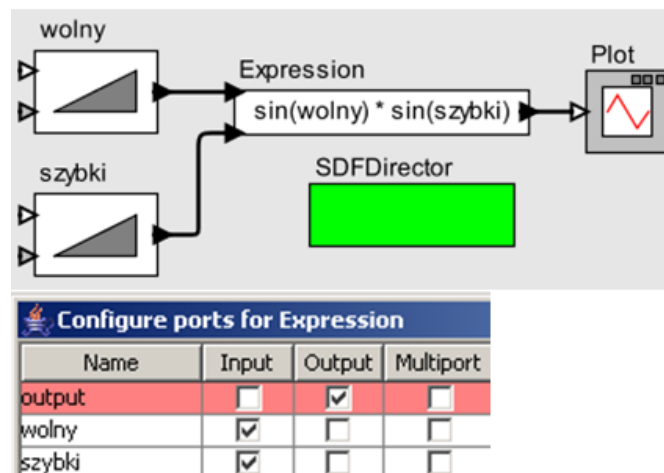
Symulacje z rysunku 6. lub 8., po dodaniu do nich jednostronnego limitera -- **Limiter2** i **Accumulator'a**, można wykorzystać do zademonstrowania pracy układu demodulatora sygnału zmodulowanego częstotliwościowo (FM), co przedstawiono na rys.10. W rzeczywistym demodulatorze FM accumulator jest układem stratnym. Wyniki uzyskanej symulacji demodulatora z rys. 10. zaprezentowano na rys. 11.



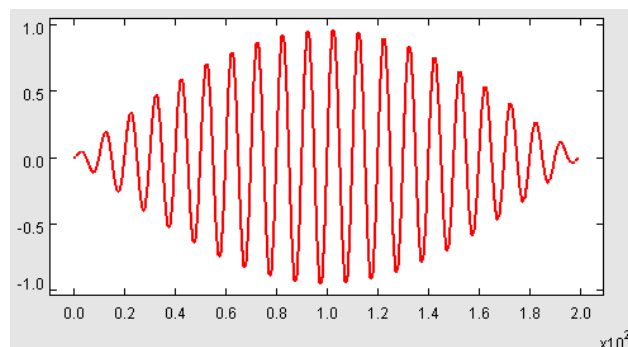
Rys. 11. Wynik symulacyjne demodulatora FM z rysunku 10.

3.3. Actor Expression w symulowaniu algorytmów

Expression jest ważnym i bardzo użytecznym aktorem. Można go zastosować wówczas, jeśli symulowany algorytm lub jego część daje się opisać wyrażeniem analitycznym. Nie ma przy tym ograniczeń stopnia złożoności funkcji opisującej symulowany algorytm. Prosty przykład dydaktyczny dla zastosowania **Expression** może być układ przedstawiony na rys.12, którego wyniki symulacji prezentuje rys.13.



Rys. 12. Przykład struktury modelu układu zawierającego Expression z pokazaną poprawną konfiguracją portów.



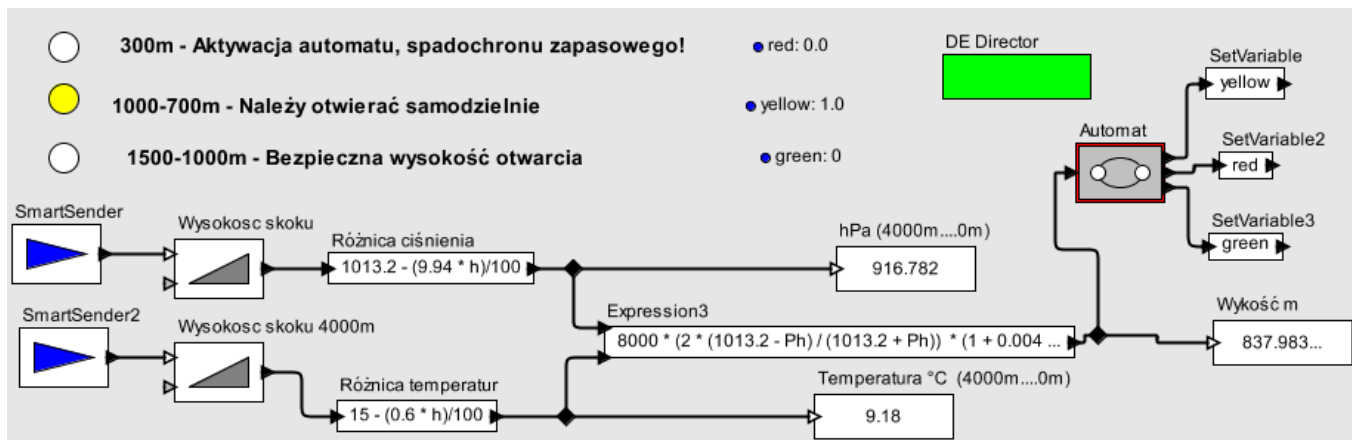
Rys. 13. Wyniki symulacji układu z rys 12,

4. SYMULACJA ALGORYTMU W ŚRODOWISKU PTOLEMY II w DOMENIE DISCRETE EVENTS

Podczas edycji modeli symulacyjnych dla innych domen nie ma zasadniczych różnic w zbiorach dostępnych aktorów. Podstawowa zmiana polega na przydzieleniu do projektu požądanego **Director** tak, jak to było widoczne w poprzednim rozdziale przy tworzeniu projektów symulacyjnych dla domeny SDF. Dokonany wybór jednoznacznie określa domenę systemu PTOLEMY II w której będzie przeprowadzana aktualna symulacja - w tym przypadku Discrete Events (DE).

Rysunek 14. przedstawia strukturę symulowanego projektu nazwanego roboczo -- *bezpieczny spadochron (BS)*. W symulowanym BS przyjęto 4000m za wysokość z której skacze skoczek. W projekcie wykorzystano zmianę ciśnienia atmosferycznego i temperatury podczas opadania skoczka. Na poziomie morza przy temperaturze 15°C ciśnienie

atmosferyczne wynosi 1013,2hPa, zaś na wysokości 4000m 616,5hPa. Temperatura powietrza zmniejsza się o 0,6°C na każde 100m wznoszenia. W **Expression1 (Różnica ciśnienia)** wykorzystano zależność: $1013.2 - (9.94 * h)/100$, natomiast w **Różnica temperatur: 15 - (0.6 * h)/100** **Expression3** ma postać: $8000 * (2 * (1013.2 - Ph) / (1013.2 + Ph)) * (1 + 0.004 * ((15 + Th) / 2))$, gdzie: Ph - ciśnienie atmosferyczne, a Th - temperatura. W symulowanym systemie odpowiednie stany związane z odległością skoczka od ziemi są sygnalizowane poprzez zapalenie się jednej z trzech lampek (zielonej, żółtej bądź czerwonej). Układ ten po zaimplementowaniu w mikrokontrolerze i dołączeniu do niego wymaganych czujników będzie sygnalizował/sterował odpowiednimi elementami wykonawczymi po przekroczeniu określonych wysokości opadania skoczka spadochronowego.



Rys. 14. Struktura symulowanego systemu *bezpieczny spadochron*.

5. WNIOSKI KOŃCOWE

Przedstawione symulacje pokazują tylko niektóre możliwości systemu PTOLEMY II.

Obserwowane podczas symulacji określone charakterystyki i sygnały wyjściowe badanych systemów/algorytmów pozwalają na ich ocenę jakościową.

Celem przeprowadzonych symulacji było przyspieszenie poprawnej implementacji hardware'owej.

System PTOLEMY II może być także wykorzystany do symulacji w wielu innych dziedzinach, takich jak: chemia, badania konstrukcji budowlanych oraz ekonomia nie wykluczając również prognoz giełdowych.

Zastosowanie modelu **Expression** w wielu sytuacjach ułatwia/upraszcza tworzenie symulowanej struktury algorytmu/podsystemu. Często udaje się wykorzystać kilka modeli Expression (rys. 14.), co dodatkowo przyspiesza opracowanie projektu sumulacyjnego opisanego nie tylko jednym wyrażeniem matematycznym.

6. BIBLIOGRAFIA

1. <http://ptolemy.berkeley.edu/ptolemyII/>
2. W. Sześciński, J. Majewski. Ptolemy Software Environment in Designing and Simulation of Digital Signal Processing Circuits, The Scientific Papers of Electrical Engineering Faculty, Computer Aided Education. Technical University of Gdańsk, Gdańsk, Vol. 8, 1995, pp. 9-18.

SIMULATION VERIFICATION OF ALGORITHMS IN PTOLEMY II ENVIRONMENT

Key-words: embedded systems, Ptolemy system, algorithms simulation

Summary The paper presents Ptolemy II domain. Selected Synchronous Data Flow domain capabilities applicable to verification of algorithms using simulation methods are presented in detail. Positive simulation result can determine the implementation of the simulated algorithm, for example, in embedded systems. Presented solution allows to speed-up the process of creating the prototype, and leads to lowering development costs.