



AGH

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

DZIEDZINA NAUK INŻYNIERYJNO-TECHNICZNYCH

AUTOMATYKA, ELEKTRONIKA, ELEKTROTECHNIKA I TECHNOLOGIE KOSMICZNE

ROZPRAWA DOKTORSKA

Zastosowanie rachunku różniczkowego niecałkowitego rzędu w modelowaniu cyfrowym procesów przewodnictwa cieplnego.

Autor: mgr inż. Edyta Gawin

Promotor rozprawy: prof. dr hab. inż. Krzysztof Oprzędkiewicz

Praca wykonana: Akademia Górniczo-Hutnicza
Wydział Elektrotechniki, Automatyki, Informatyki
i Inżynierii Biomedycznej,
Katedra Automatyki i Robotyki

Kraków, 2023



AGH

AGH UNIVERSITY OF KRAKOW

ENGINEERING AND TECHNOLOGY

AUTOMATION, ELECTRONICS, ELECTRICAL ENGINEERING AND SPACE
TECHNOLOGIES

DOCTORAL THESIS

Application of non-integer order calculus in digital modelling
of heat transfer processes.

Author: Edyta Gawin

First supervisor: Prof. Krzysztof Oprzedkiewicz

Completed in: AGH University of Krakow
Faculty of Electrical Engineering, Automatics, Computer Science
and Biomedical Engineering,
Department of Automatic Control and Robotics

Krakow, 2023

Pragnę złożyć serdeczne podziękowania promotorowi Panu Profesorowi Krzysztofowi Oprędkiewiczowi za trud włożony w opiekę naukową oraz za wszelką pomoc przy powstawaniu tej pracy.

Dziękuję Rodzinie za wsparcie i cierpliwość.

Streszczenie

Praca zawiera opracowanie, analizę teoretyczną, identyfikację oraz weryfikację doświadczalną dla nowych, zaproponowanych przez autorkę modeli niecałkowitego rzędu dla procesów przewodnictwa cieplnego. Porównano cztery modele ciągle zdefiniowane w przestrzeni stanu: pierwszy klasyczny, rzędu całkowitego oraz trzy nowe modele niecałkowitego rzędu. Wyniki przeprowadzonych eksperymentów wskazały, iż najlepszym modelem pod względem minimalizacji funkcji kosztu MSE, jest model o niecałkowitym rzędzie zarówno wzdłuż długości, jak i czasu. Następnie przedstawiono nową, efektywną obliczeniowo dyskretną metodę rozwiązywania równań niecałkowitego rzędu, bazującą na aproksymacji CFE. Zaproponowano również dyskretne modele w przestrzeni stanu wykorzystujące aproksymacje dyskretne FOBD oraz CFE. Opracowano i zweryfikowano doświadczalnie nowe modele transmitancyjne niecałkowitego rzędu w postaci hybrydowych transmitancji ciągłych i dyskretnych.

W ostatniej części pracy dokonano implementacji na sterowniku PLC podstawowego elementu ułamkowego rzędu z użyciem dyskretnych aproksymacji FOBD i CFE. Badania testowe dyskretnych implementacji PLC podstawowego operatora ułamkowego wykazały, że może on być zaimplementowany na platformie PLC ze spełnieniem wymagań dotyczących zarówno dokładności, jak i wymagań czasu rzeczywistego.

Abstract

The dissertation includes development, theoretical analysis, identification and experimental verification of new, proposed by the author, models of fractional order of heat transfer processes. Four continuous models defined in the state space were compared: the first classical model, of integer order, and three new models of fractional order. The results of the conducted experiments indicated that the best model in terms of minimizing the MSE cost function is the model with incomplete order both along length and time. Next, a new computationally efficient discrete method for solving fractional order equations based on the CFE approximation is presented. Discrete state-space models using discrete FOBD and CFE approximations have also been proposed. New fractional order transfer function models in the form of hybrid continuous and discrete transfer functions were implementation and experimentally verified.

In the last part of the work, the basic fractional order element was implemented on the PLC using discrete FOBD and CFE approximations. Test studies of discrete PLC implementations of the basic fractional operator have shown that it can be implemented on a PLC platform meeting both accuracy and real-time requirements.

Spis treści

Spis rysunków	12
Spis tabel	14
Uwagi wstępne	15
Tezy i zadania naukowe zrealizowane w pracy	16
Zawartość pracy	17
1. Preliminaria	19
1.1. Podstawowe pojęcia i definicje.....	19
1.2. Modele systemów dynamicznych: transmitancja i równanie stanu niecałkowi- tego rzędu	27
1.3. Aproksymacje ciągłe i dyskretne operatorów niecałkowitego rzędu	29
1.3.1. Aproksymacja Oustaloupa (ORA).....	29
1.3.2. Aproksymacja Charefa.....	30
1.3.3. Aproksymacja PSE	31
1.3.4. Aproksymacja CFE.....	33
2. Rozważany doświadczalny obiekt cieplny i jego modele całkowitego rzędu	35
2.1. Konstrukcja obiektu (w uproszczeniu).....	35
2.2. Model całkowitego rzędu w przestrzeni stanu	39
2.3. Modele zastępcze całkowitego rzędu w postaci transmitancji	42
3. Modele niecałkowitego rzędu dla rozważanego obiektu	45
3.1. Ciągłe modele niecałkowitego rzędu w przestrzeni stanu.....	45
3.1.1. Ciągły model ułamkowego rzędu względem czasu oraz długości w przestrzeni stanu.....	46
3.1.2. Uprozczone modele ułamkowego rzędu w przestrzeni stanu.....	54
3.1.3. Porównanie oraz identyfikacja wartości parametrów ciągłych modeli w przestrzeni stanu.....	60

3.2. Dyskretny model niecałkowitego rzędu w przestrzeni stanu	67
3.3. Ciągły model w postaci transmitancji hybrydowej niecałkowitego rzędu	78
3.4. Dyskretny model transmitancji niecałkowitego rzędu z opóźnieniem.....	81
4. Implementacja PLC podstawowego operatora ułamkowego rzędu.....	87
4.1. Ogólne uwagi na temat implementacji specjalnych algorytmów sterowania na platformie PLC	88
4.2. Systemy doświadczalne użyte w badaniach.....	89
4.2.1. Sterownik PLC SIEMENS 1200.....	89
4.2.2. Sterownik PLC SIEMENS 1500.....	90
4.3. Implementacja operatora na platformie SIEMENS SIMATIC.....	91
4.4. Testy spełnienia wymagań czasu rzeczywistego podczas realizacji obliczeń ułamkowych	94
4.5. Testy dokładności i zbieżności	97
Podsumowanie.....	107
Dodatek A: Kody źródłowe oprogramowania w Matlabie.....	109
Dodatek B: Kody źródłowe oprogramowania PLC	117
Dodatek C: Algorytm kukułki	187
Bibliografia	196

Spis rysunków

1.1	Funkcja Gamma Eulera	19
1.2	Jednoparametrowa funkcja Mittag-Lefflera (dla $\alpha < 1$)	21
1.3	Jednoparametrowa funkcja Mittag-Lefflera (dla $\alpha > 1$)	21
1.4	Wykres operatora różniczki niecałkowitego rzędu (GL) po czasie dla funkcji stałej ($\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$)	23
1.5	Wykres operatora różniczki niecałkowitego rzędu (GL) po czasie dla funkcji stałej ($\alpha = 1.2, 1.5, 1.8, 2.2, 2.5$)	24
1.6	System dynamiczny ułamkowego rzędu, gdzie $u(t)$ - wejście systemu, $y(t)$ - wyjście systemu	27
2.1	Konstrukcja systemu eksperymentalnego z jednym wejściem i trzema wyjściami	36
2.2	Uproszczony schemat obiektu doświadczalnego z jednym wejściem i trzema wyjściami	36
2.3	Charakterystyka statyczna eksperymentalnego obiektu cieplnego z jednym wejściem i trzema wyjściami	37
2.4	Rozkład czasowo-przestrzenny odpowiedzi skokowej rozpatrywanego obiektu .	38
2.5	Odpowiedzi skokowe dla wszystkich wyjść obiektu	39
3.1	Funkcja kosztu (3.18) jako funkcja wartości rzędu N	51
3.2	Czasowo-przestrzenny rozkład temperatury dla modelu całkowitego rzędu i $N = 15$	52
3.3	Czasowo-przestrzenny rozkład temperatury dla modelu niecałkowitego rzędu i $N = 15$	52
3.4	Błąd modelu całkowitego rzędu w funkcji czasu i długości pręta dla $N = 15$.	53
3.5	Błąd modelu niecałkowitego rzędu w funkcji czasu i długości pręta dla $N = 15$	53
3.6	Błąd dla modeli z rzędem całkowitym (.) oraz niecałkowitym (+), rząd $N = 15$.	59
3.7	Błąd dla modeli z rzędem całkowitym (.) oraz niecałkowitym (+), rząd $N = 25$.	59

3.8	Odpowiedź skokowa modelu rzeczywistego obiektu oraz modelu niecałkowitego rzędu dla $N = 25$	60
3.9	Zależność wartości funkcji kosztu MSE od rzędu N dla wszystkich testowanych modeli	63
3.10	Odpowiedź skokowa obiektu oraz modelu 1 z rzędem całkowitym ($\alpha = 1, \beta = 2$) oraz optymalnym rzędem $N = 10, a = 0.000806, R_a = 0.034214$	63
3.11	Odpowiedź skokowa obiektu oraz modelu 2 z rzędem niecałkowitym ($\alpha = 0.931555, \beta = 2$) oraz optymalnym rzędem $N = 22, a = 0.000419, R_a = 0.066532$	64
3.12	Odpowiedź skokowa obiektu oraz modelu 3 z rzędem niecałkowitym ($\alpha = 1, \beta = 2.089926$) oraz optymalnym rzędem $N = 14, a = 0.000646, R_a = 0.034243$	64
3.13	Odpowiedź skokowa obiektu oraz modelu 4 z rzędem niecałkowitym ($\alpha = 0.930289, \beta = 1.998808$) oraz optymalnym rzędem $N = 22, a = 0.000410, R_a = 0.0677066$	65
3.14	Widma układów stabilnych: $h=1$ s, $N=20$	74
3.15	Widma układów niestabilnych: $h=1$ s, $N=25$	75
3.16	Zależność funkcji kosztu MSE (3.34) dla różnych rzędów N i wszystkich testowanych modeli	77
3.17	Odpowiedź skokowa obiektu oraz modelu z niecałkowitym rzędem ($\alpha = 0.9319, \beta = 2.0306$) oraz optymalnego rzędu $N = 18, L = 100, a = 0.000561, R_a = 0.0548$	77
3.18	Odpowiedź skokowa: obiektu (linia ciągła), modelu (3.61) z $n_1 = 2$ (oznaczone jako "+") oraz modelu (3.61) z $n_1 = 3$ (oznaczone ".")	80
3.19	Odpowiedź skokowa: obiektu (linia ciągła), modelu (3.62) z $n_2 = 2$ (oznaczone jako "+") oraz modelu (3.62) z $n_2 = 3$ (oznaczone ".")	80
3.20	Odpowiedź skokowa obiektu oraz modelu (3.65)	84
3.21	Odpowiedź skokowa obiektu oraz modelu (3.66)	85
4.1	Konstrukcja systemu doświadczalnego ze sterownikiem SIEMENS S7 1200	90
4.2	Konstrukcja systemu doświadczalnego ze sterownikiem SIEMENS S7 1500	90
4.3	Diagram LD zastosowany do testów	95
4.4	Zależność funkcji kosztu MSE i czasu trwania od długości pamięci L dla $\alpha = 0, 5$ w aproksymacji PSE	99

4.5	Zależność funkcji kosztu MSE i czasu trwania od długości pamięci L dla $\alpha = -0,5$ w aproksymacji PSE	100
4.6	Zależność funkcji kosztu MSE i czasu trwania od długości pamięci L dla $\alpha = 0,5$ w aproksymacji CFE	102
4.7	Zależność funkcji kosztu MSE i czasu trwania od długości pamięci L dla $\alpha = -0,5$ w aproksymacji PSE	102
4.8	Odpowiedzi skokowe aproksymacji PSE i CFE w porównaniu z odpowiedzią analityczną dla $\alpha = 0,5$	103
4.9	Odpowiedzi skokowe aproksymacji PSE i CFE w porównaniu z odpowiedzią analityczną dla $\alpha = -0,5$	103

Spis tabel

1.1	Współczynniki aproksymacji CFE $CFE_{N,D}(z^{-1}, \alpha)$ dla aproksymacji Tustina według [54]	34
2.1	Parametry modelu w przestrzeni stanu	43
2.2	Parametry modeli transmitacyjnych całkowitego rzędu	44
3.1	Funkcja kosztu (3.18) dla modelu całkowitego oraz niecałkowitego rzędu oraz różnych wartości N	50
3.2	Optymalne wartości parametrów modelu całkowitego rzędu dla różnych N	51
3.3	Optymalne wartości parametrów modelu niecałkowitego rzędu dla różnych N	51
3.4	Parametry modeli całkowitego i ułamkowego rzędu oraz wartość funkcji kosztu (3.32).	58
3.5	Funkcja kosztu MSE (3.34) dla różnych N oraz wszystkich testowanych modeli	62
3.6	Parametry obiektu cieplnego	73
3.7	Maksymalny wymiar modelu N dla różnych okresów próbkowania h	73
3.8	Maksymalny wymiar okresu próbkowania h dla ustalonego wymiaru modelu N	74
3.9	Funkcja kosztu MSE (3.34) dla różnych rzędów N i wszystkich testowanych modeli	76
3.10	Parametry modelu (3.61)	79
3.11	Parametry modelu (3.62)	79
3.12	Porównanie modeli całkowitego oraz ułamkowego rzędu	81
3.13	Optymalne parametry i funkcja kosztów MSE (3.69) dla modelu (3.65) oraz dla modelu całkowitego rzędu (2.14)	83
3.14	Optymalne parametry i funkcja kosztów MSE (3.69) dla modelu (3.66) oraz dla modelu całkowitego rzędu (2.14)	83
4.1	Komponenty oprogramowania dla aproksymacji PSE	92
4.2	Komponenty oprogramowania dla aproksymacji CFE	93

4.3	Zmienne sterownika PLC SIEMENS 1200 / 1500	94
4.4	Zmienne panelu HMI	94
4.5	Czasy wykonywania funkcji PSEcoeff (obliczanie współczynników d_l dla aproksymacji PSE) w [ms]	96
4.6	Czasy wykonywania bloku funkcyjnego instancji PSE (obliczenie aproksymacji PSE) w OB30 w [ms]	96
4.7	Czasy wykonywania funkcji CFEEcoeff (obliczenie współczynników w_m i v_m dla aproksymacji CFE w [ms]	96
4.8	Czasy wykonywania instancji bloku funkcyjnego CFE (obliczanie aproksymacji CFE) w OB30 w [ms]	97
4.9	Funkcja kosztu MSE (4.1) dla aproksymacji PSE dla dodatnich wartości α . .	98
4.10	Funkcja kosztu MSE (4.1) dla aproksymacji PSE dla ujemnych wartości α . .	99
4.11	Funkcja kosztu MSE (4.1) dla aproksymacji CFE dla dodatnich wartości α . .	101
4.12	Funkcja kosztu MSE (4.1) dla aproksymacji CFE dla ujemnych wartości α . .	101
4.13	Odpowiedź skokowa: obiektu (czerwona linia) oraz modelu z różnymi wartościami rzędu α oraz różnymi rzędami L dla aproksymacji PSE	104
4.14	Odpowiedź skokowa: obiektu (czerwona linia) oraz modelu z różnymi wartościami rzędu α oraz różnymi rzędami M dla aproksymacji CFE	105
15	Optymalne parametry i funkcja kosztów MSE (3.69) dla modelu (3.65) oraz dla modelu całkowitego rzędu (2.14) otrzymane z użyciem algorytmu kukułki . . .	187
16	Optymalne parametry i funkcja kosztów MSE (3.69) dla modelu (3.65) oraz dla modelu całkowitego rzędu (2.14) otrzymane z użyciem algorytmu kukułki . . .	188

Uwagi wstępne

Teoria równań różniczkowych niecałkowitego rzędu została wprowadzona przez matematyków XVII wieku (Gottfried Wilhelm Leibniz (1646-1716), Guillaume de l'Hospital (1661-1704), Isaac Newton (1643-1727), Leonhard Euler (1707-1783) czy Pierre Simon de Laplace (1749-1827)), ale jej zastosowanie było ówczasie nieznane. Od tego czasu rachunek różniczkowy rzędu ułamkowego był przedmiotem rozważań teoretycznych wielu pokoleń matematyków.

Dalsza ewolucja teorii i zastosowań nastąpiła dopiero w XIX i XX wieku. Joseph Liouville, Georg Wilhelm Grünwald, Ilia Letnikov, Bernhard Riemann oraz Michele Caputo pod koniec XIX wieku stworzyli podstawy rachunku różniczkowego ułamkowego (niecałkowitego) rzędu.

W XX wieku zauważono, iż wiele zjawisk fizycznych można lepiej opisać używając rachunku różniczkowego ułamkowego rzędu. Dodatkowo bardzo szybki rozwój technik komputerowych spowodował, że pojawiło się coraz więcej praktycznych zastosowań. Rachunek różniczkowy rzędu ułamkowego może być traktowany jako uogólnienie pojęć całki i różniczki w klasycznym rachunku całkowym i różniczkowym.

Modelowanie procesów i zjawisk trudnych do opisanania za pomocą innych struktur matematycznych jest jednym z głównych obszarów zastosowań rachunku niecałkowitego rzędu. Obecnie znane są szerokie zastosowania rachunku niecałkowitego rzędu, m.in. - w automatyce: modelowanie procesów o trudnej do zdefiniowania dynamice (wymieniki ciepła, masy, dyfuzja, superkondensatory, zachowanie się polimerów, ruchy Browna, modelowanie fraktalne), regulatory niecałkowitego rzędu ($FOPI^\lambda D^\mu$) - [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11],[12], [13],[14]. Analizę problemu dyfuzji anomalnej z wykorzystaniem podejścia ułamkowego rzędu i teorii półgrup przedstawił np. [15]. Zastosowanie równań różniczkowych niecałkowitego rzędu może być interpretowane jako uogólnienie znanych modeli całkowitego rzędu.

Powszechnie wiadomo, że procesy wymiany ciepła można również modelować z wykorzystaniem podejścia niecałkowitego rzędu. Problemem tym zajmowali się m.in. [16], [8], [17]. Należy zauważyć, że wszystkie znane modele mają postać transmitancji lub równania różniczkowego cząstkowego. Model przestrzeni stanów niecałkowitego rzędu dla procesu wymiany

ciepła nie został jeszcze przedstawiony. W niniejszej rozprawie przedstawiono propozycję nowego, przestrzennego modelu procesu wymiany ciepła w ośrodku jednowymiarowym. Idea tego modelu opiera się bezpośrednio na modelu półgrupowym dla jednowymiarowego problemu wymiany ciepła.

Tezy i zadania naukowe zrealizowane w pracy

Tezy rozważanej rozprawy można zdefiniować następująco:

1. Jest możliwe podanie dokładnego opisu procesu przewodnictwa cieplnego w postaci ciągłego w czasie, nieskończonego wymiarowego równania stanu niecałkowitego rzędu. Model taki spełnia założenie o dekompozycji widma i jego skończenie wymiarowa aproksymacja może być użyta do modelowania procesu cieplnego z użyciem np. środowiska Matlab.
2. Poprawny opis procesu cieplnego może być także dokonany z użyciem modeli zastępczych w postaci transmitancji hybrydowych, zawierających zarówno część całkowitego rzędu, jak i część rzędu ułamkowego.
3. W przypadku rozważanej klasy systemów rozważane modele ciągłe w czasie mogą zostać przekształcone do postaci modeli dyskretnych o złożoności obliczeniowej umożliwiającej ich implementację na platformie typowego sterownika przemysłowego, np. sterownika PLC z użyciem znormalizowanych narzędzi programistycznych opisanych przez standard 61131.3.

Celem realizacji zdefiniowanych tez rozprawy zaplanowano wykonanie następujących zadań naukowych:

1. Opracowanie, analiza teoretyczna i weryfikacja doświadczalna ciągłych modeli niecałkowitego rzędu w przestrzeni stanu opisujących rozważany proces przewodnictwa cieplnego. Modele te powinny bazować na dobrze znanych wynikach i być ich uogólnieniem niecałkowitego rzędu.
2. Opracowanie nowej, efektywnej obliczeniowo dyskretnej metody rozwiązywania równań stanu niecałkowitego rzędu, bazującej na aproksymacji CFE.
3. Opracowanie, analiza teoretyczna i weryfikacja doświadczalna modeli dyskretnych w przestrzeni stanu, wykorzystujących aproksymacje dyskretne: FOBD i CFE.

4. Opracowanie i weryfikacja doświadczalna nowych modeli transmitancyjnych niecałkowitego rzędu dla rozważanego procesu przewodnictwa cieplnego. Modele te mają mieć postać hybrydowych (zawierających część całkowitego i niecałkowitego rzędu) transmitancji ciągłych i dyskretnych.
5. Implementacja na sterowniku PLC podstawowego elementu ułamkowego z użyciem aproksymacji FOBD i CFE oraz testy dokładności i szybkości działania tych aproksymacji.

Zawartość pracy

Praca składa się z czterech zasadniczych rozdziałów, wstępu, podsumowania oraz trzech dodatków.

W uwagach wstępnych został podany rys historyczny oraz przykłady obecnych zastosowań rachunku niecałkowitego rzędu w automatyce - modelowanie procesów o trudnej do zdefiniowania dynamice czy też regulatory niecałkowitego rzędu.

Rozdział pierwszy zawiera podstawowe pojęcia i definicje z zakresu rachunku różniczkowego niecałkowitego rzędu: funkcja Gamma oraz jedno i dwuargumentowa funkcja Mittag-Lefflera, operator różniczko-całki po czasie wg definicji GL, RL i C, operator różniczko-całki po długości wg Riesz, dyskretna pochodna ułamkowa po czasie, operatory Caputo-Fabrizio, Atangana-Baleanu-Caputo, Atangana-Baleanu-Riemanna, transmitancja niecałkowitego rzędu, równanie stanu niecałkowitego rzędu i jego rozwiązanie. Zostały tu też podane i omówione aproksymacje podstawowych elementów niecałkowitego rzędu: ciągłe Oustaloupa (ORA) i Charefa oraz dyskretne PSE i CFE.

W rozdziale drugim omówiono konstrukcję obiektu doświadczalnego oraz jego znany model półgrupowy całkowitego rzędu. Zostało podane klasyczne równanie różniczkowe procesu przewodnictwa cieplnego w ośrodku jednowymiarowym i jego modele dokładne i zastępcze całkowitego rzędu: modele zastępcze Kupfmuellera oraz Strejca w postaci transmitancji, równanie różniczkowe ciepłoprzewodnictwa oraz jego zapis w postaci równania stanu w przestrzeni Hilberta z modelem półgrupowym.

W kolejnym rozdziale podano i omówiono proponowane modele niecałkowitego rzędu i ich elementarne własności. Są to modele niecałkowitego rzędu ciągłe w czasie zdefiniowane w przestrzeni stanu, będące uogólnieniem modelu półgrupowego całkowitego rzędu, oraz modele zastępcze w postaci transmitancji hybrydowej niecałkowitego rzędu. Przeprowadzono analizę podstawowych własności modeli częściowo niecałkowitego rzędu oraz podano rozwiązania równania stanu dla tych modeli. Podane zostały także modele niecałkowitego rzędu dyskretne

w czasie, dla których zbadano optymalny rząd oraz okres próbkowania. Zostały podane warunki konieczne i wystarczające praktycznej stabilności układu dyskretnego. Podane są tu również skończone wymiarowe aproksymacje ciągłe i dyskretne modeli niecałkowitego rzędu. W tym rozdziale przedstawiono porównanie dokładności oraz zbieżności wszystkich rozważanych modeli oraz przeprowadzono identyfikację parametrów badanych modeli całkowitego oraz niecałkowitego rzędu.

W ostatnim rozdziale zaproponowane modele są weryfikowane z wykorzystaniem wyników eksperymentów na platformie sprzętowo-programowej sterownika PLC. Podano konstrukcje układów eksperymentalnych na platformie PLC wykorzystujące rodzinę SIEMENS S7 (PLC SIEMENS SIMATIC S7 1200 oraz PLC SIEMENS SIMATIC S7 1500). Implementacja aproksymacji omawianych modeli dyskretnych na platformie sterownika PLC została wykonana zgodnie z podejściem obiektowym zalecanym w normie 61131.3. Została przeprowadzona optymalizacja rzędu i identyfikacja parametrów rozważanych modeli w przestrzeni stanu z wykorzystaniem funkcji kosztu MSE. Przeprowadzono testy dokładności, szybkości działania oraz spełnienia wymagań czasu rzeczywistego w implementacji na platformie PLC.

Wszystkie podane modele niecałkowitego rzędu dla procesów przewodnictwa cieplnego można zastosować do zaimplementowania na platformie PLC. W szczególności można je użyć do:

1. sterowania bazującego na modelowaniu (Model Based Control) - przedstawione w [18],
2. sterowania bazującego na modelu diagnostycznym uszkodzeń (Model Based Fault Detection - MBFD).

Badania, których wyniki przedstawiono w niniejszej rozprawie, autorka niniejszej rozprawy, wraz ze współautorami opublikowała w pracach: [19], [20], [21], [22], [23], [24], [25], [26],[27], [28], [29],[30],[31], [32],[33],[34].

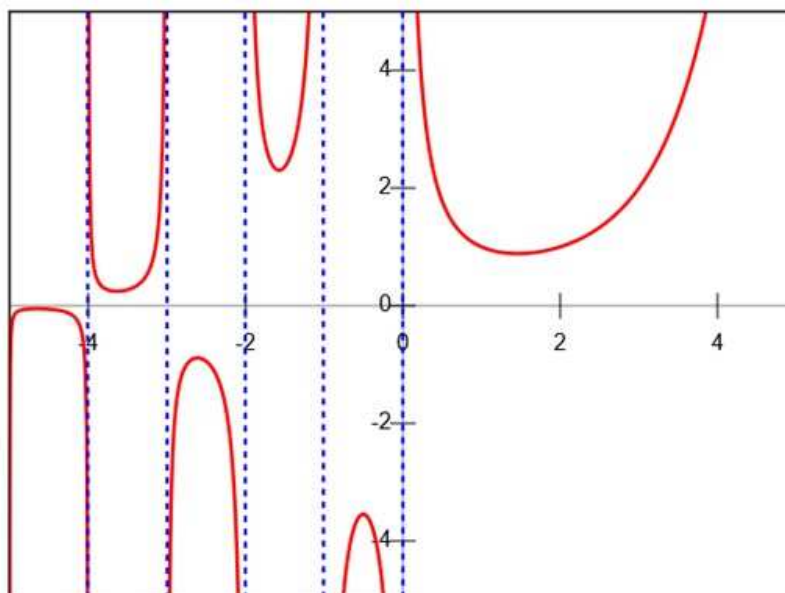
1. Preliminaria

1.1. Podstawowe pojęcia i definicje

Zestaw podstawowych pojęć i definicji rozpoczniemy od przypomnienia definicji funkcji Gamma Eulera (np. [35]):

Definicja 1. *Funkcja Gamma Eulera (kompletna)*

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt \quad \text{dla } x \in \mathbb{Z} \quad (1.1)$$



Rys. 1.1. Funkcja Gamma Eulera

Przebieg kompletnej funkcji Gamma jest pokazany na Rys. 1.1. Funkcja Gamma Eulera rozszerza pojęcie silni na zbiór liczb rzeczywistych i zespolonych:

$$\Gamma(x + 1) = x\Gamma(x) \quad (1.2)$$

Istnieją także niekompletne funkcje Gamma Eulera zdefiniowane jak poniżej:

Definicja 2. *Funkcje Gamma Eulera (niekompletne)*

$$\begin{aligned}\gamma(x, \tau) &= \int_0^{\tau} t^{x-1} e^{-t} dt \quad \text{dla } x \in \mathbb{Z} \\ \Gamma(x, \tau) &= \int_{\tau}^{\infty} t^{x-1} e^{-t} dt \quad \text{dla } x \in \mathbb{Z}\end{aligned}\tag{1.3}$$

Oczywiście

$$\gamma(x, \tau) + \Gamma(x, \tau) = \Gamma(x).\tag{1.4}$$

Następnie zostanie wprowadzona funkcja Mittag-Lefflera. Jest to uogólnienie niecałkowitego rzędu funkcji $e^{\lambda t}$ i odgrywa kluczową rolę w rozwiązaniu równań stanu niecałkowitego rzędu. Jednoparametrowa funkcja Mittag-Lefflera zdefiniowana jest jako:

Definicja 3. *Jednoparametrowa funkcja Mittag-Lefflera*

$$E_{\alpha}(x) = \sum_{k=0}^{\infty} \frac{x^k}{\Gamma(k\alpha + 1)}\tag{1.5}$$

Przykładowy przebieg funkcji $E_{\alpha}(x)$ pokazany jest na Rys. 1.2 (dla $\alpha < 1$) oraz na Rys. 1.3 (dla $\alpha > 1$).

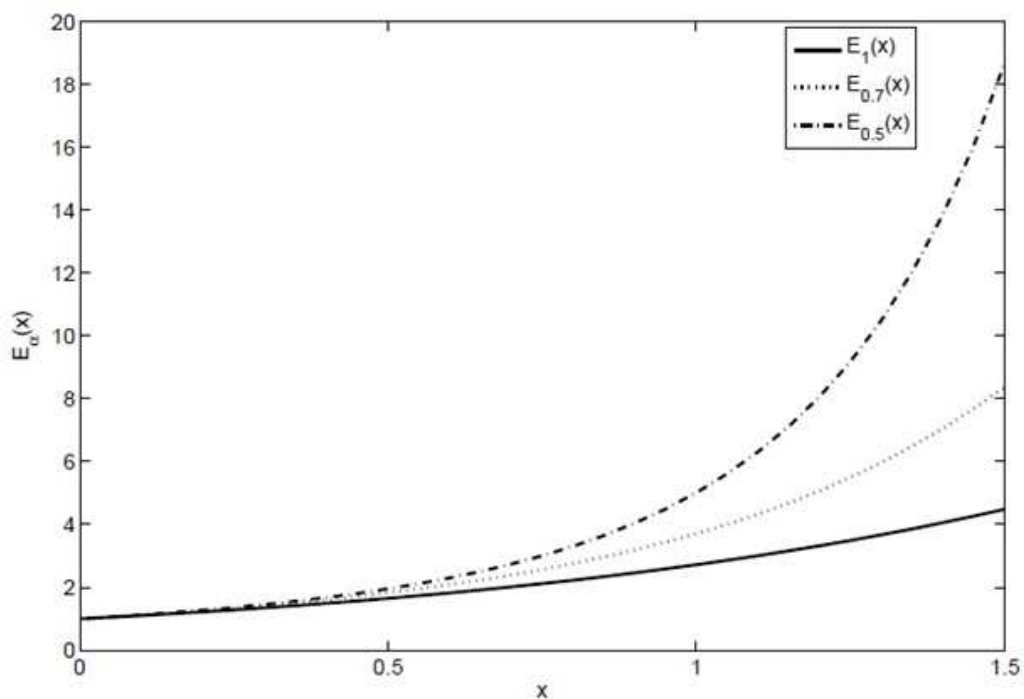
Dwuparametrowa funkcja Mittag-Lefflera określona jest następująco:

Definicja 4. *Dwuparametrowa funkcja Mittag-Lefflera*

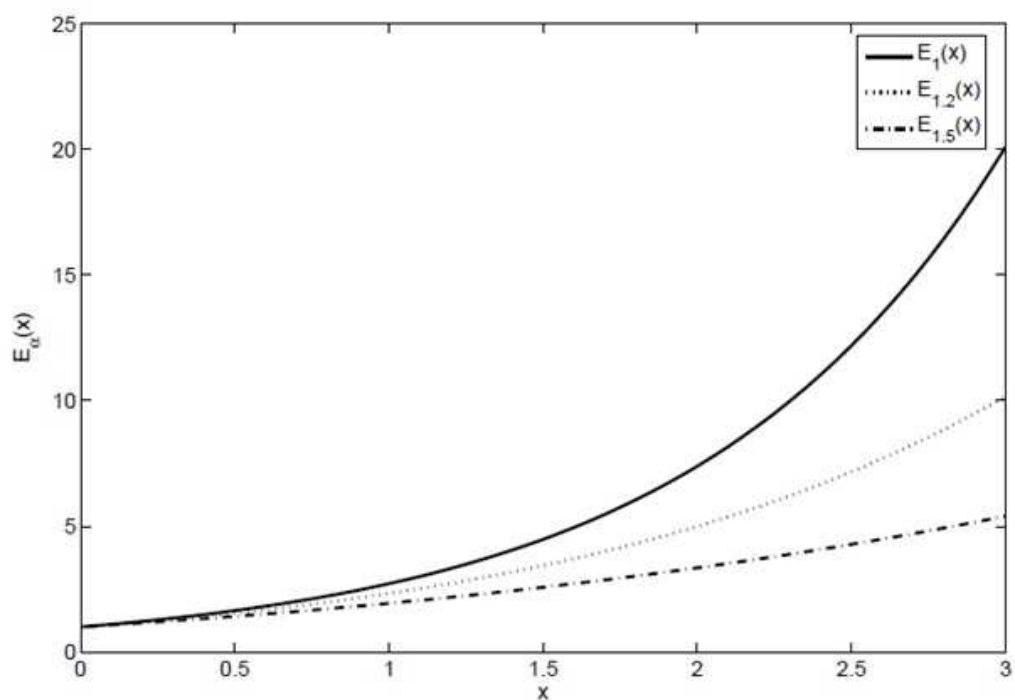
$$E_{\alpha, \beta}(x) = \sum_{k=0}^{\infty} \frac{x^k}{\Gamma(k\alpha + \beta)}\tag{1.6}$$

Dla $\beta = 1$ dwuparametrowa funkcja (1.6) sprowadza się do jednoparametrowej funkcji (1.5).

Teraz zdefiniujemy operator różniczko-całki niecałkowitego rzędu. Wyraża się on w następujący sposób (patrz np. [12]):



Rys. 1.2. Jednoparametrowa funkcja Mittag-Lefflera (dla $\alpha < 1$)



Rys. 1.3. Jednoparametrowa funkcja Mittag-Lefflera (dla $\alpha > 1$)

Definicja 5. Operator różniczko-całki niecałkowitego rzędu po czasie

$${}_a D_t^\alpha f(t) = \begin{cases} \frac{d^\alpha f(t)}{dt^\alpha}, & \alpha > 0 \\ 1, & \alpha = 0 \\ \int_a^t f(\tau)(d\tau)^{-\alpha}, & \alpha < 0 \end{cases}. \quad (1.7)$$

gdzie a oraz t oznaczają granice czasowe do obliczenia operatora (obliczenia są dla przedziału, a nie w punkcie), $\alpha \in \mathbb{R}$ niecałkowity rząd operatora. Często przyjmujemy $a = 0$.

Operator różniczkowo-całkowy ułamkowego rzędu może być opisany przez różne definicje. Najczęściej są stosowane trzy definicje podane przez: Grünwalda i Letnikova (GL), Riemanna i Liouville'a (RL) oraz Caputo (C). Wszystkie te definicje podano poniżej. Przy pewnych dodatkowych założeniach definicje te są równoważne.

Definicja 6. Definicja Grünwalda-Letnikova operatora ułamkowego rzędu (zobacz [6],[36])

$${}^{GL} D_t^\alpha f(t) = \lim_{h \rightarrow 0} h^{-\alpha} \sum_{j=0}^{\lfloor \frac{t}{h} \rfloor} (-1)^j \binom{\alpha}{j} f(t - jh) \quad (1.8)$$

gdzie $\lfloor \frac{t}{h} \rfloor$ oznacza część całkowitą z wyrażenia.

We wzorze (1.8) $\binom{\alpha}{j}$ jest uogólnieniem symbolu Newtona na zbiór liczb rzeczywistych:

$$\binom{\alpha}{j} = \begin{cases} 1, & j = 0 \\ \frac{\alpha(\alpha-1)\dots(\alpha-j+1)}{j!}, & j > 0 \end{cases} \quad (1.9)$$

Definicja 7. Definicja Riemanna-Liouville'a operatora ułamkowego rzędu

$${}^{RL} D_t^\alpha f(t) = \frac{1}{\Gamma(N - \alpha)} \frac{d^N}{dt^N} \int_0^\infty (t - \tau)^{N-\alpha-1} f(\tau) d\tau \quad (1.10)$$

gdzie $N - 1 < \alpha < N$ opisuje operator niecałkowitego rzędu, Γ jest zespoloną kompletną funkcją Gamma Eulera zdefiniowaną przez (1.1).

Definicja Caputo jest opisana jak poniżej:

Definicja 8. Definicja Caputo operatora ułamkowego rzędu (zobacz [37])

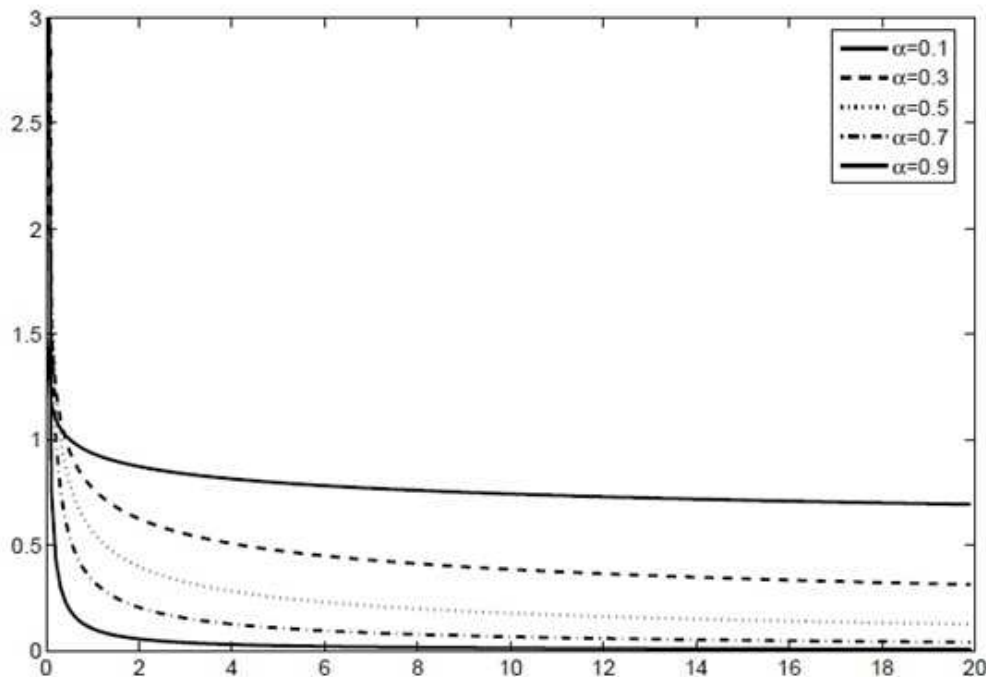
$${}^C D_t^\alpha f(t) = \frac{1}{\Gamma(N - \alpha)} \int_0^\infty \frac{f^{(N)}(\tau)}{(t - \tau)^{\alpha+1-N}} d\tau \quad (1.11)$$

Pomiędzy powyższymi definicjami istnieje następujący związek (zobacz [4]):

Niech $\alpha \leq 0, n = \lceil \alpha \rceil$ oraz funkcja f jest n razy różniczkowalna na przedziale $[a, b]$, wtedy

$${}^GL_a D_t^\alpha f(t) = T_{n-1}(f, a)(t) + {}^C_a D_t^\alpha f(t) = {}^{RL}_a D_t^\alpha f(t)$$

gdzie $T_{n-1}(f, a)(t)$ oznacza rozwinięcie funkcji f w wielomian Taylora stopnia $n-1$ w punkcie a .



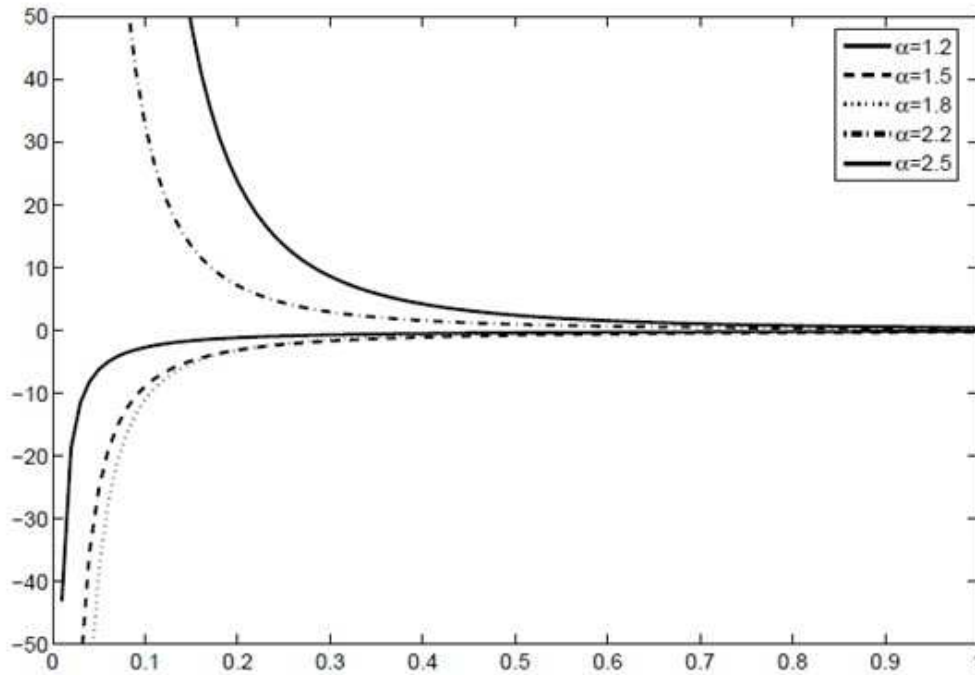
Rys. 1.4. Wykres operatora różniczki niecałkowitego rzędu (GL) po czasie dla funkcji stałej ($\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$)

Przykładowy wykres operatora różniczki niecałkowitego rzędu (GL) po czasie dla funkcji stałej podany jest na Rys. 1.4 (dla $\alpha < 1$) oraz na Rys. 1.5 (dla $\alpha > 1$).

Dla operatorów RL oraz C transformata Laplace'a zdefiniowana jest jako (zobacz [12]):

Definicja 9. Transformata Laplace'a dla operatora Riemann - Liouville'a

$$\begin{aligned} \mathcal{L}\{{}^{RL}_0 D_t^\alpha f(t)\}(s) &= s^\alpha F(s), \quad \alpha < 0 \\ \mathcal{L}\{{}^{RL}_0 D_t^\alpha f(t)\}(s) &= s^\alpha F(s) - \sum_{k=0}^{n-1} s^k {}_0 D_t^{\alpha-k-1} f(0), \\ &\alpha > 0, \quad n-1 < \alpha \leq n \in \mathbb{N} \end{aligned} \quad (1.12)$$



Rys. 1.5. Wykres operatora różniczkowego niecałkowitego rzędu (GL) po czasie dla funkcji stałej ($\alpha = 1.2, 1.5, 1.8, 2.2, 2.5$)

Definicja 10. Transformata Laplace'a dla operatora Caputo

$$\begin{aligned} \mathcal{L}\{ {}_0^C D_t^\alpha f(t) \}(s) &= s^\alpha F(s), \quad \alpha < 0 \\ \mathcal{L}\{ {}_0^C D_t^\alpha f(t) \}(s) &= s^\alpha F(s) - \sum_{k=0}^{n-1} s^{\alpha-k-1} {}_0 D_t^k f(0), \\ \alpha > 0, \quad n-1 < \alpha \leq n \in \mathbb{N} \end{aligned} \quad (1.13)$$

Konsekwentnie, odwrotna transformata Laplace'a dla operatorów niecałkowitego rzędu dana jest następująco (zobacz np. [35]):

$$\begin{aligned} \mathcal{L}^{-1}\{s^\alpha F(s)\} &= {}_0 D_t^\alpha f(t) + \sum_{k=0}^{n-1} \frac{t^{k-1}}{\Gamma(k-\alpha+1)} f^{(k)}(0^+) \\ n-1 < \alpha < n, \quad n \in \mathbb{Z} \end{aligned} \quad (1.14)$$

Dla elementu s^α za pomocą wzoru (1.14) postać analityczna odpowiedzi skokowej wyrażona jest jak poniżej:

$$y_{an}(t) = \frac{t^{-\alpha}}{\Gamma(1-\alpha)}. \quad (1.15)$$

Powyższy analityczny wzór odpowiedzi skokowej (1.15) będzie interpretowany jako wzorek do oszacowania dokładności testowanych implementacji PLC.

W ostatnich latach pojawiły się też inne definicje operatorów niecałkowitego rzędu, jak np. operatory Caputo-Fabrizio (CF) ([38]) czy też Atangana-Baleanu (AB) ([39]).

Operator Caputo-Fabrizio został podany po raz pierwszy w [38], następnie został także omówiony w [40]. Operator wynika ze wzoru (1.11).

Definicja 11. *Definicja Caputo-Fabrizio operatora ułamkowego rzędu*

$${}_0^{CF}D_t^\alpha f(t) = \frac{M_n(\alpha)}{(1-\alpha)} \int_0^\infty \dot{f}(\tau) \exp\left(\frac{\alpha(t-\tau)}{(1-\alpha)}\right) \tau \quad (1.16)$$

gdzie $M_n(0) = M_n(1) = 1$ jest funkcją normującą.

Dla operatora Caputo-Fabrizio można również podać transformatę Laplace'a (zobacz np. [38]). Dla $0 \leq \alpha < 1$ przyjmuje postać:

Definicja 12. *Transformata Laplace'a dla operatora Caputo-Fabrizio*

$$\mathcal{L}\{{}_0^{CF}D_t^\alpha f(t)\}(s) = \frac{s\mathcal{L}(f(t) - f(0))}{s + \alpha(1-s)} \quad (1.17)$$

Operator Atangana-Baleanu został podany w [39], następnie został także omówiony w [41], oraz [42]. Operator Atangana-Baleanu pochodnej rzędu ułamkowego uzyskuje się przez zastąpienie jądra wykładniczego w operatorze Caputo-Fabrizio (CF) przez jądro Mittag-Lefflera. Operator AB zdefiniowany jest za pomocą definicji C lub RL pochodnej rzędu ułamkowego. Korzystając z tych definicji otrzymujemy odpowiednio operator Atangana-Baleanu-Caputo (ABC) lub Atangana-Baleanu-Riemanna (ABR):

Definicja 13. *Definicja Atangana-Baleanu-Caputo operatora ułamkowego rzędu*

$${}_a^{ABC}D_t^\alpha f(t) = M_\alpha \int_a^t f'(x) E_\alpha\left(-\alpha \frac{(t-x)^\alpha}{1-\alpha}\right) dx \quad (1.18)$$

gdzie $E_\alpha(\dots)$ jest jednoparametrową funkcją Mittag-Lefflera, M_α jest funkcją normującą równą:

$$M_\alpha = 1 - \alpha + \frac{\alpha}{\Gamma(\alpha)}$$

oraz $\Gamma(\dots)$ jest funkcją Gamma Eulera (1.1).

Definicja 14. *Definicja Atangana-Baleanu-Riemann operatora ułamkowego rzędu*

$${}_a^{ABR}D_t^\alpha f(t) = M_\alpha \frac{d}{dt} \int_a^t f(x) E_\alpha(-\alpha \frac{(t-x)^\alpha}{1-\alpha}) dx \quad (1.19)$$

gdzie $E_\alpha(\dots)$ jest jednoparametrową funkcją Mittag-Lefflera, M_α jest funkcją normującą równą:

$$M_\alpha = 1 - \alpha + \frac{\alpha}{\Gamma(\alpha)}$$

oraz $\Gamma(\dots)$ jest funkcją Gamma Eulera (1.1).

Transformaty Laplace'a dla pochodnych ABC i ABR są następujące:

Definicja 15. *Transformata Laplace'a dla operatora Atangana-Baleanu-Caputo*

$$\mathcal{L}\{{}_a^{ABC}D_t^\alpha f(t)\}(s) = \frac{M_\alpha}{1-\alpha} \frac{s^\alpha \{f(t)\}(s) - s^{\alpha-1} f(0)}{s^\alpha + \frac{\alpha}{1-\alpha}} \quad (1.20)$$

Definicja 16. *Transformata Laplace'a dla operatora Atangana-Baleanu-Riemanna*

$$\mathcal{L}\{{}_a^{ABR}D_t^\alpha f(t)\}(s) = \frac{M_\alpha}{1-\alpha} \frac{s^\alpha \{f(t)\}(s)}{s^\alpha + \frac{\alpha}{1-\alpha}} \quad (1.21)$$

Dla homogenicznego warunku początkowego: $f(0) = 0$ obie transformaty Laplace'a są równe:

$$\mathcal{L}\{{}_a^{ABC}D_t^\alpha f(t)\}(s) = \mathcal{L}\{{}_a^{ABR}D_t^\alpha f(t)\}(s). \quad (1.22)$$

Przejdziemy do przypomnienia definicji pochodnej Riesz. Pochodna przestrzenna niecałkowitego rzędu podana przez Riesz ma następującą postać (patrz np. [43]):

Definicja 17. *Definicja pochodnej przestrzennej Riesz niecałkowitego rzędu*

$$\frac{\partial^\gamma \Theta(x, t)}{\partial x^\gamma} = -d_\gamma ({}_0D_x^\gamma + {}_x D_1^\gamma) \Theta(x, t) \quad (1.23)$$

gdzie:

$$d_\gamma = \frac{1}{2 \cos(\frac{\pi\gamma}{2})}, \quad \text{dla } \gamma \neq 1 \quad (1.24)$$

W (1.23) ${}_0D_x^\gamma$ oraz ${}_xD_1^\gamma$ oznaczają pochodne lewo- i prawo- stronne Riemanna-Liouville zdefiniowane następująco:

$${}_0D_x^\gamma = \frac{1}{\Gamma(2-\alpha)} \frac{\partial}{\partial x} \int_0^x \frac{\Theta(\xi, t) d\xi}{(x-\xi)^{\gamma-1}} \quad (1.25)$$

$${}_xD_1^\gamma = \frac{1}{\Gamma(2-\alpha)} \frac{\partial}{\partial x} \int_x^1 \frac{\Theta(\xi, t)}{(\xi-x)^{\gamma-1}} \quad (1.26)$$

W (1.25) oraz (1.26) $\Gamma(\cdot)$ oznacza funkcję Gamma wprowadzoną wzorem (1.1).

1.2. Modele systemów dynamicznych: transmitancja i równanie stanu niecałkowitego rzędu

Transmitancja stosująca przekształcenie Laplace'a: $G(s) = \int_0^\infty g(t)e^{-st} dt$ jest bardzo użytecznym modelem wielu systemów dynamicznych. W przypadku układów ułamkowych transmitancja ta jest funkcją zmiennej zespolonej s^α gdzie $\alpha \in \mathbb{R}$. Użyteczność tego modelu w Matlabie jest określona przez możliwość jego modelowania z wykorzystaniem skończonej wymiarowych aproksymacji rzędu całkowitego.

Idea transmitancji systemu niecałkowitego rzędu jest analogiczna do idei transmitancji dla systemów dynamicznych całkowitego rzędu.



Rys. 1.6. System dynamiczny ułamkowego rzędu, gdzie $u(t)$ - wejście systemu, $y(t)$ - wyjście systemu

Rozważmy układ dynamiczny pokazany na rysunku 1.6. Liniowe równanie różniczkowe ułamkowego rzędu opisujące zależność pomiędzy wejściem i wyjściem tego układu ma następującą postać:

$$a_n D^{\alpha_n} y(t) + \dots + a_1 D^{\alpha_1} y(t) + a_0 D^{\alpha_0} y(t) = b_m D^{\beta_m} u(t) + \dots + b_1 D^{\beta_1} u(t) + b_0 D^{\beta_0} u(t) \quad (1.27)$$

gdzie $\alpha_n, \dots, \alpha_0, \beta_m, \dots, \beta_0$ - ułamkowe rzędy pochodnej, $a_n, \dots, a_0, b_m, \dots, b_0$ - współczynniki równań różniczkowych.

Niech $U(s)$ i $Y(s)$ będą transformatami Laplace'a sygnałów wejściowych i wyjściowych, a wszystkie warunki początkowe będą równe zero. Wtedy transmitancję ułamkowego rzędu powyższego systemu można wyrazić analogicznie, jak w przypadku systemu całkowitego rzędu:

$$G(s) = \frac{Y(s)}{U(s)} \quad (1.28)$$

gdzie $Y(s) = \mathcal{L}(y(t))$ oraz $U(s) = \mathcal{L}(u(t))$.

Ułamkowa transmitancja ma postać:

$$G(s) = \frac{b_m s^{\beta_m} + \dots + b_1 s^{\beta_1} + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + \dots + a_1 s^{\alpha_1} + a_0 s^{\alpha_0}} \quad (1.29)$$

przy założeniu zerowych warunków początkowych.

Liniowe, stacjonarne równanie stanu dla układu liniowego ułamkowego rzędu ma postać (zobacz [44],[45],[46]):

$$\begin{aligned} \frac{d^\alpha x(t)}{dt^\alpha} &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \quad (1.30)$$

gdzie $x(t) \in \mathbb{R}^N$ jest wektorem stanu, $u(t) \in \mathbb{R}^P$ jest wektorem sterowania, a $y(t) \in \mathbb{R}^R$ jest wektorem wyjścia, $0 < \alpha < 1$ jest ułamkowym rzędem równania stanu.

Podany został przypadek równania z tym samym rzędem ułamkowym dla wszystkich zmiennych stanu. Można również rozważać równania, gdzie rzędy ułamkowe poszczególnych równań "składowych", są różne.

Jeżeli operator ułamkowego rzędu jest opisany przy użyciu definicji Caputo (1.11), to wtedy rozwiązanie liniowego równania stanu (1.30) przyjmuje postać (zobacz [35]):

$$x(t) = \Phi_0(t)x_0 + \int_0^t \Phi(t-\tau)Bu(\tau)d\tau \quad (1.31)$$

gdzie:

$$\Phi_0(t) = E_\alpha(At^\alpha) = \sum_{k=0}^{\infty} \frac{A^k t^{k\alpha}}{\Gamma(k\alpha + 1)} \quad (1.32)$$

$$\Phi(t) = \sum_{k=0}^{\infty} \frac{A^k t^{(k+1)\alpha-1}}{\Gamma((k+1)\alpha)} \quad (1.33)$$

1.3. Aproksymacje ciągłe i dyskretne operatorów niecałkowitego rzędu

Aproksymacje całkowitego rzędu układów ułamkowych zostały przedstawione przez wielu autorów, na przykład: [47], [48], [49], [50]. Modelowanie transmitancji ułamkowego rzędu w środowisku Matlab / Simulink wymaga zastosowania modelu skończonego wymiarowego całkowitego rzędu. W niniejszej pracy zostaną przedstawione aproksymacje zaproponowane przez Oustaloupa i Charefa, aproksymujące oba obiekty elementarne opisane przez transmitancje s^α i $\frac{1}{(Ts+1)^\alpha}$. Obie aproksymacje są bardzo podobne i ich ogólny zamysł polega na dopasowaniu wykresu Bodego modułu amplitudy aproksymacji do wykresu Bodego modułu amplitudy aproksymowanej transmitancji. Dla obiektu o rozważanych transmitancjach s^α i $\frac{1}{(Ts+1)^\alpha}$ możliwe jest wyznaczenie dokładnej charakterystyki częstotliwościowej przy użyciu znanych metod.

1.3.1. Aproksymacja Oustaloupa (ORA)

Aproksymacja Oustaloupa (zobacz np. [51], [52], [53]) dedykowana jest do modelowania elementarnych transmitancji ułamkowego rzędu w postaci:

$$G(s) = s^{\alpha_1} \quad (1.34)$$

gdzie $\alpha_1 \in \mathbb{R}$.

Aproksymacja Oustaloupa transmitancji (1.34) jest zdefiniowana następująco:

$$G_{Ost} = \frac{\prod_{n=1}^M \left(1 + \frac{s}{\mu_n}\right)}{\prod_{n=1}^M \left(1 + \frac{s}{\nu_n}\right)} \quad (1.35)$$

W (1.35) M oznacza rząd aproksymacji, μ i ν oznacza odpowiednio zera i bieguny aproksymacji, które można obliczyć następująco:

$$\mu_1 = \omega_l \sqrt{\eta} \quad (1.36)$$

$$\nu_n = \mu_n \gamma \quad n = 1, \dots, M \quad (1.37)$$

$$\mu_{n+1} = \nu_n \eta \quad n = 1, \dots, M - 1 \quad (1.38)$$

gdzie:

$$\gamma = \left(\frac{\omega_h}{\omega_l}\right)^{\left(\frac{\alpha_1}{M}\right)} \quad (1.39)$$

$$\eta = \left(\frac{\omega_h}{\omega_l}\right)^{\left(\frac{1-\alpha_1}{M}\right)} \quad (1.40)$$

W (1.39) i (1.40) ω_l i ω_h oznacza zakres pulsacji, dla których aproksymacja jest zdefiniowana.

Jeżeli zachodzi konieczność zapewnienia zbieżności odpowiedzi skokowej aproksymacji z odpowiedzią skokową rzeczywistego obiektu w stanie ustalonym, to dodatkowo we wzorze (1.35) musi pojawić się wzmocnienie k_f obliczane jako dodatkowy opcjonalny parametr:

$$G_{Ost} = k_f \frac{\prod_{n=1}^M (1 + \frac{s}{\mu_n})}{\prod_{n=1}^M (1 + \frac{s}{\nu_n})} \quad (1.41)$$

1.3.2. Aproksymacja Charefa

Aproksymacja Charefa (zobacz [47]) dedykowana jest do modelowania elementarnych transmitancji ułamkowego rzędu opisanych przez element inercyjny:

$$G(s) = \frac{1}{(T_{\alpha_2} s + 1)^{\alpha_2}} \quad (1.42)$$

gdzie T oznacza stałą czasową obiektu, $0 < \alpha_2 < 1$ oznacza ułamkowy rząd obiektu.

Zauważmy, że obie podstawowe transmitancje podane powyżej (poprzez wzory (1.34) oraz (1.42)) można zastosować do opisanego znacznie bardziej skomplikowanych obiektów. Dodatkowo bardzo interesujący model można zbudować jako kombinację transmitancji całkowitego i ułamkowego rzędu. Modele takie zaproponowane i opracowane przy współdziałaniu autorki są omówione w dalszej części pracy.

Aproksymacja zaproponowana przez Charefa pozwala nam na przybliżenie obiektu inercyjnego ułamkowego rzędu rozważanego w niniejszej pracy (wzór (1.42)).

Aproksymacja Charefa transmitancji (1.42) jest zdefiniowana następująco:

$$G_{Cha}(s) = \frac{\prod_{i=0}^N (1 + \frac{s}{z_i})}{\prod_{i=0}^N (1 + \frac{s}{p_i})} \quad (1.43)$$

gdzie z_i oraz p_i oznaczają odpowiednio zera i bieguny aproksymacji, N oznacza rząd aproksymacji. Aproksymowana transmitancja jest bardzo podobna do (1.35).

Założeniem tej aproksymacji jest jak najlepiej wpasować wykres charakterystyk Bodego aproksymacji do wykresu charakterystyk Bodego modelu w danym paśmie częstotliwości. Zera

i bieguny są obliczane przy użyciu następujących zależności rekursywnych:

$$p = \frac{1}{T} \quad (1.44)$$

$$p_0 = p\sqrt{b} \quad (1.45)$$

$$p_i = p_0(ab)^i \quad i = 1, \dots, N \quad (1.46)$$

$$z_i = ap_0(ab)^i \quad i = 1, \dots, N \quad (1.47)$$

gdzie:

$$a = 10^{\frac{\Delta}{10(1-\alpha_1)}} \quad (1.48)$$

$$b = 10^{\frac{\Delta}{10\alpha_1}} \quad (1.49)$$

W (1.48) i (1.49) Δ oznacza maksymalny błąd dopuszczalny aproksymacji Charefa, zdefiniowany jako różnica między modulem charakterystyki częstotliwościowej modelu i obiektu, wyrażony w [dB].

1.3.3. Aproksymacja PSE

Dyskretna aproksymacja PSE (Power Series Expansion) wynika bezpośrednio z definicji Grünwalda-Letnikova (wzór (1.8)).

Jej dyskretna wersja nazywa się Fractional Order Backward Difference (FOBD) dla $\alpha > 0$ i analogicznie Fractional Order Backward SUM (FOBS) dla $\alpha < 0$:

Definicja 18. *Dyskretna pochodna ułamkowa po czasie*

$$(\Delta^\alpha x)(t) = \frac{1}{h^\alpha} \sum_{l=0}^{\infty} (-1)^l \binom{\alpha}{l} x(t - lh). \quad (1.50)$$

Oznaczmy współczynnik $(-1)^l \binom{\alpha}{l}$ przez d_l :

$$d_l = (-1)^l \binom{\alpha}{l}. \quad (1.51)$$

Współczynnik (1.51) można również obliczyć za pomocą następującej, równoważnej formuły rekurencyjnej (patrz np. [6]), przydatnej podczas obliczeń numerycznych:

$$d_0 = 1$$

$$d_l = \left(1 - \frac{1 + \alpha}{l}\right) d_{l-1}, \quad l = 1, \dots, L. \quad (1.52)$$

W rzeczywistości obliczenie sumy (1.50) jest możliwe tylko dla skończonych wartości l . Maksymalna wartość l jest znana jako "długość pamięci" L , a skończona aproksymacja (1.50) przyjmuje następującą postać:

Definicja 19. *Różnica wsteczna ułamkowego rzędu*

$$(\Delta^\alpha x)(t) = \frac{1}{h^\alpha} \sum_{l=0}^L d_l x(t - lh). \quad (1.53)$$

W (1.53) L oznacza długość pamięci konieczną do poprawnej aproksymacji operatora niecałkowitego rzędu. Niestety dobra zgodność przybliżenia PSE wymaga użycia długiej pamięci L co może utrudniać cyfrową implementację.

FOBD (1.53) można również wyrazić jako dyskretny filtr FIR zawierający wyłącznie zera:

$$(\Delta^\alpha x)(t) = \frac{1}{h^\alpha} \sum_{l=0}^L d_l z^{-l}. \quad (1.54)$$

Odpowiedź czasowa na dyskretne wejście $u^+(t)$ ma postać:

$$y_{PSE}^+(k) = \frac{1}{h^\alpha} \sum_{l=0}^L d_l u^+(k - l). \quad (1.55)$$

gdzie $y_{PSE}^+(k)$ jest wyjściem w k -tym kroku, $u^+(k - l)$ oznacza sygnały wejściowe w $k - l$ -tym momencie czasu, d_l są współczynnikami aproksymacji PSE, wyznaczonymi przez (1.52).

Implementacja aproksymacji FOBD na sterowniku PLC jest to klasyczna i intuicyjna metoda cyfrowego modelowania elementów FO, ponieważ wynika bezpośrednio z definicji GL operatora FO. Z drugiej strony głównym problemem podczas jego cyfrowej implementacji jest „długość pamięci” niezbędna do zachowania rozsądnej dokładności. Wyraża się ona długością tablicy statycznej użytej do zapisania poprzednich wartości obliczonej pochodnej. Oczywiście rzeczywisty rozmiar używanej pamięci statycznej może być znacznie większy. Spowodowane jest to tym, że jego rzeczywista wielkość wymaga uwzględnienia również liczby wykonanych operacji FO (np. pojedyncza pętla sterowania FO PID wymagała od nas użycia 2 instancji operatora FO) oraz zamodelowanego elementu dynamicznego: rzędu transmitancji lub wymiaru równania stanu.

Równanie (1.55) zostanie bezpośrednio zaimplementowane jako blok funkcyjny (FB) w sterowniku PLC. Użycie FB jest spowodowane tym, że poprawne obliczenie (1.55) wymaga znajomości L poprzednich kroków wyjścia, a FB jest najmniejszą Jednostką Organizacji Programu (POU - Program Organization Block) zapewniającą „funkcję pamięci” dla jego zmiennych. Niestety wartość L zapewniająca rozsądną dokładność tego przybliżenia musi być długa (zazwyczaj większa niż 100). Fakt ten może powodować problemy podczas implementacji w czasie rzeczywistym i należy go przetestować, aby uniknąć błędów wynikających z ograniczeń czasowych. Wyniki takich testów zostaną podane w rozdziale 4.2.

1.3.4. Aproksymacja CFE

Aproksymacja CFE jest alternatywnym podejściem do dyskretnego modelowania operatora FO. Ma ona postać filtra IIR - zawiera zarówno bieguny, jak i zera. Jest szybka i łatwa w realizacji, ponieważ jej rząd jest zwykle stosunkowo niski (najczęściej nie wyższy niż 5).

Dyskretyzację elementu ułamkowego s^α , $\alpha \in \mathbb{R}$ można wykonać za pomocą tak zwanej funkcji generującej $s \approx \omega(z^{-1})$. Nowy operator podniesiony do potęgi α ma następującą postać (zobacz np. [54], [55]):

$$\begin{aligned} (\omega(z^{-1}))^\alpha &= \left(\frac{1+a}{h}\right)^\alpha CFE\left\{\left(\frac{1-z^{-1}}{1+az^{-1}}\right)^\alpha\right\}_{M,M} = \\ &= \frac{P_{\alpha M}(z^{-1})}{Q_{\alpha M}(z^{-1})} = \left(\frac{1+a}{h}\right)^\alpha \frac{CFE_N(z^{-1}, \alpha)}{CFE_D(z^{-1}, \alpha)} = \frac{\sum_{m=0}^M w_m z^{-m}}{\sum_{m=0}^M v_m z^{-m}}. \end{aligned} \quad (1.56)$$

W (1.56) a jest współczynnikiem zależnym od typu aproksymacji (na przykład: $a=0$ dla aproksymacji Eulera, $a=1$ dla aproksymacji Tustina; w pracy [56] a pojawia się jako dodatkowy parametr do optymalizacji modelu), h oznacza próbkę czasu, M jest rzędem aproksymacji. Wartości liczbowe współczynników w_m i v_m oraz różne wartości parametru a można obliczyć na przykład przy użyciu funkcji MATLAB'a podanej przez Petrasa na stronie [57] oraz [58]. Wskazana funkcja MATLAB'a została zastosowana w eksperymentach opisanych w następnych rozdziałach. Jeśli rozważamy aproksymację Tustina ($a=1$) dostajemy $CFE_D(z^{-1}, \alpha) = CFE_N(z^{-1}, -\alpha)$ oraz wielomian $CFE_D(z^{-1}, \alpha)$ można podać w bezpośredniej postaci (zobacz [54]). Przykłady wielomianów $CFE_D(z^{-1}, \alpha)$ dla $M = 1, 3, 5$ podane są w tabeli 1.1.

Odpowiedź elementu opisanego aproksymacją CFE przy pomocy wzoru (1.56) w k -tym momencie można obliczyć z wyrażenia:

$$y_{CFE}^+(k) = \frac{1}{v_0} \left[- \sum_{m=1}^M v_m y^+(k-m) + \sum_{m=0}^M w_m u^+(k-m) \right]. \quad (1.57)$$

gdzie $y_{CFE}^+(k-m)$ i $u^+(k-m)$ oznacza wyjściowy i wejściowy sygnał w $k-m$ -tym momencie odpowiednio, v_m i w_m są współczynnikami aproksymacji CFE, podanymi przykładowo w tabeli 1.1.

Równanie (1.57) zostanie bezpośrednio zaimplementowane jako blok funkcyjny (FB) w PLC. Użycie FB jest spowodowane tym, że poprawne obliczenie (1.57) wymaga znajomości M poprzednich kroków sygnałów wyjściowych i sterujących, a FB jest najmniejszą Jednostką Organizacji Programu (POU - Program Organization Block) zapewniającą „pamięć funkcyjną” dla swoich zmiennych.

Tabela 1.1. Współczynniki aproksymacji CFE $CFE_{N,D}(z^{-1}, \alpha)$ dla aproksymacji Tustina według [54]

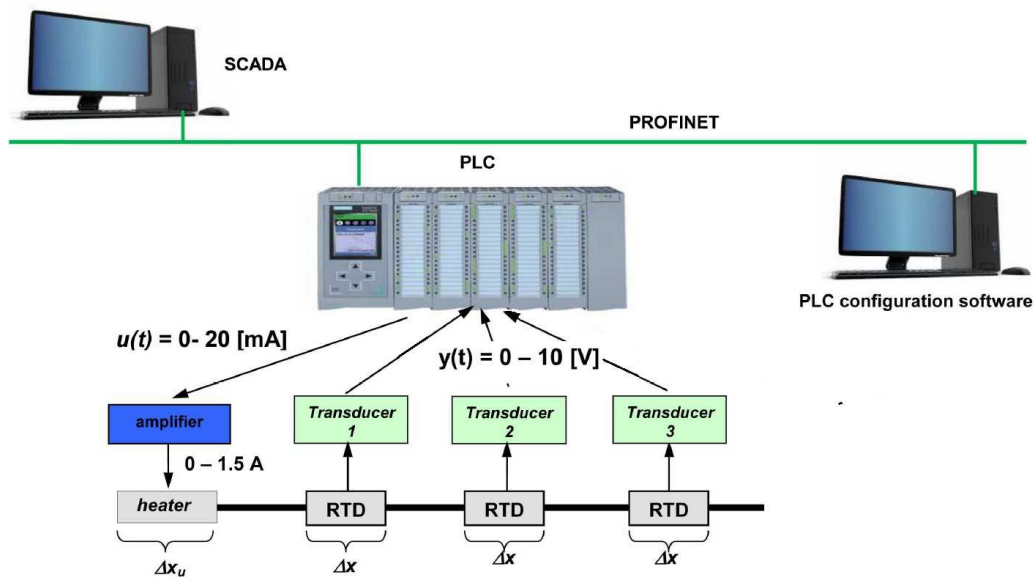
Order M	w_m	v_m
$M=1$	$w_1 = -\alpha$ $w_0 = 1$	$v_1 = \alpha$ $v_0 = 1$
$M=3$	$w_3 = -\frac{\alpha}{3}$ $w_2 = \frac{\alpha^2}{3}$ $w_1 = -\alpha$ $w_0 = 1$	$v_3 = \frac{\alpha}{3}$ $v_2 = \frac{\alpha^2}{3}$ $v_1 = \alpha$ $v_0 = 1$
$M=5$	$w_5 = -\frac{\alpha}{5}$ $w_4 = \frac{\alpha^2}{5}$ $w_3 = -\left(\frac{\alpha}{5} + \frac{2\alpha^3}{35}\right)$ $w_2 = \frac{2\alpha^2}{5}$ $w_1 = -\alpha$ $w_0 = 1$	$v_5 = \frac{\alpha}{5}$ $v_4 = \frac{\alpha^2}{5}$ $v_3 = -\left(\frac{-\alpha}{5} + \frac{-2\alpha^3}{35}\right)$ $v_2 = \frac{2\alpha^2}{5}$ $v_1 = \alpha$ $v_0 = 1$

2. Rozważany doświadczalny obiekt cieplny i jego modele całkowitego rzędu

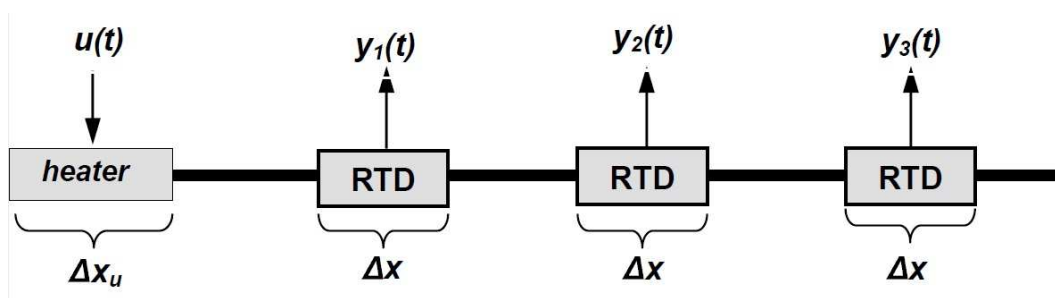
2.1. Konstrukcja obiektu (w uproszczeniu)

Systemem wykorzystywanym do badań był eksperymentalny obiekt cieplny z jednym wejściem i trzema wyjściami pokazany na rysunku 2.1 (zob. np. [59], [60], [61]). Ma on postać cienkiego miedzianego pręta o długości 260 [mm]. Do dalszych rozważań przyjmujemy, że długość pręta jest równa 1,0. Oznacza to, że w dalszych rozważaniach lokalizacja i długość grzałki oraz czujników RTD będzie wyrażana względem 1,0. Pręt jest ogrzewany za pomocą grzałki elektrycznej o długości Δx_u zlokalizowanej na jednym końcu i trzech czujników RTD typu Pt-100 o długości $\Delta x = 0,14$ znajdujących się w punktach: 0,3, 0,5 oraz 0,7 długości pręta (Rysunek 2.2.). Sygnał wejściowy systemu jest standardowym sygnałem prądowym $0 - 20[mA]$. Wzmacniany jest do zakresu od 0 do $1,5[A]$ i jest sygnałem wejściowym dla grzałki. Temperatura pręta jest mierzona za pomocą czujników Pt-100. Sygnały z czujników odczytywane są bezpośrednio przez moduł wejść analogowych w sterowniku PLC. Cały system jest połączony przez sieć PROFINET.

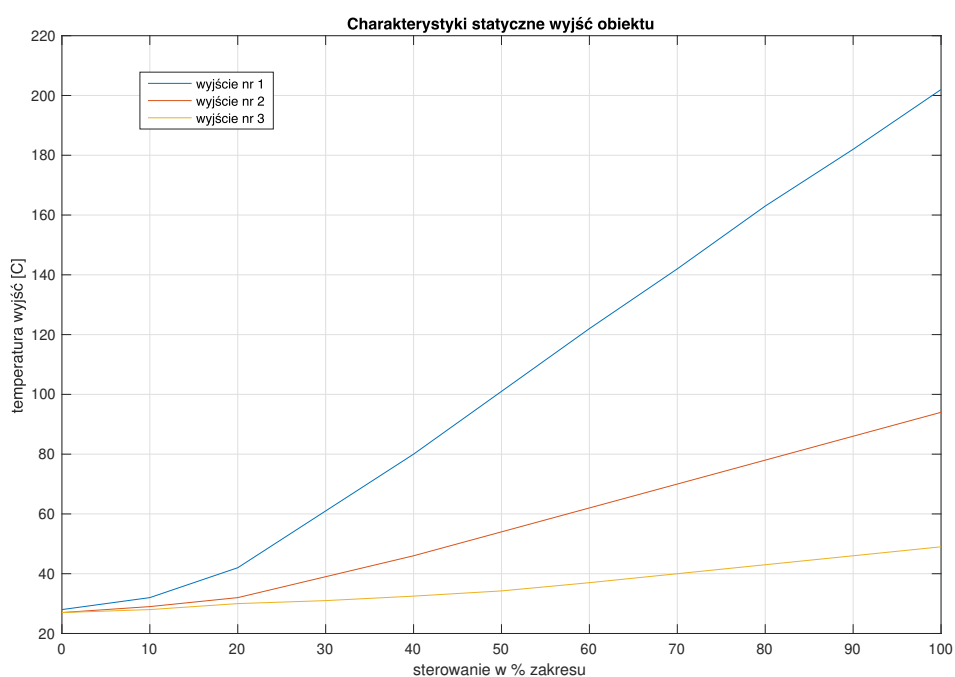
Rozpatrując charakterystykę statyczną doświadczalnego systemu czyli wykres temperatury na poszczególnych wyjściach w zależności od sterowania w procentach dopuszczalnego zakresu można wyciągnąć wniosek, iż w przedziale pomiędzy 40% a 60% zakresu sterowania zależność ta jest liniowa. Wszystkie eksperymenty prowadzone były dla punktu pracy równego 50% dopuszczalnego zakresu sterowania (w zakresie niewychodzącym poza przedział 40% i 60%), a więc możemy przyjąć, że wykorzystywany do badań obiekt cieplny jest obiektem liniowym.



Rys. 2.1. Konstrukcja systemu eksperymentalnego z jednym wejściem i trzema wyjściami

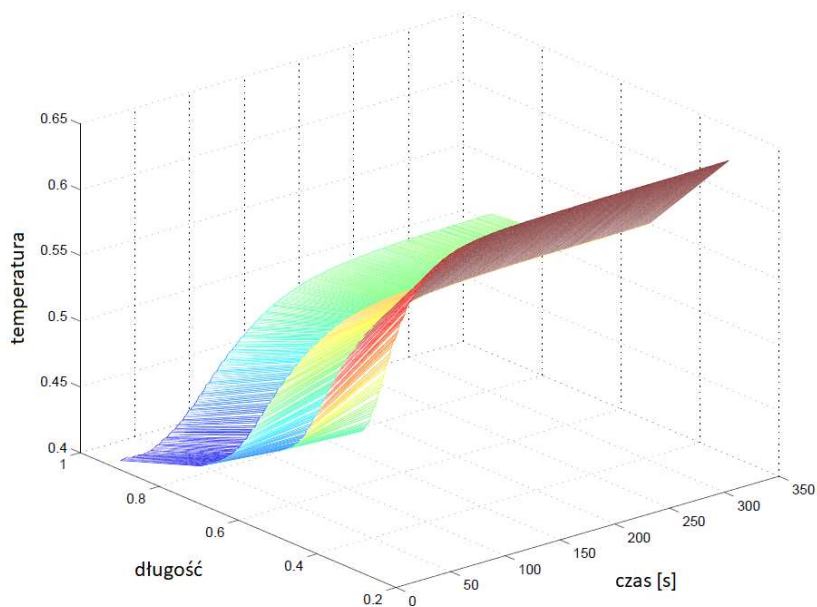


Rys. 2.2. Uproszczony schemat obiektu doświadczalnego z jednym wejściem i trzema wyjściami



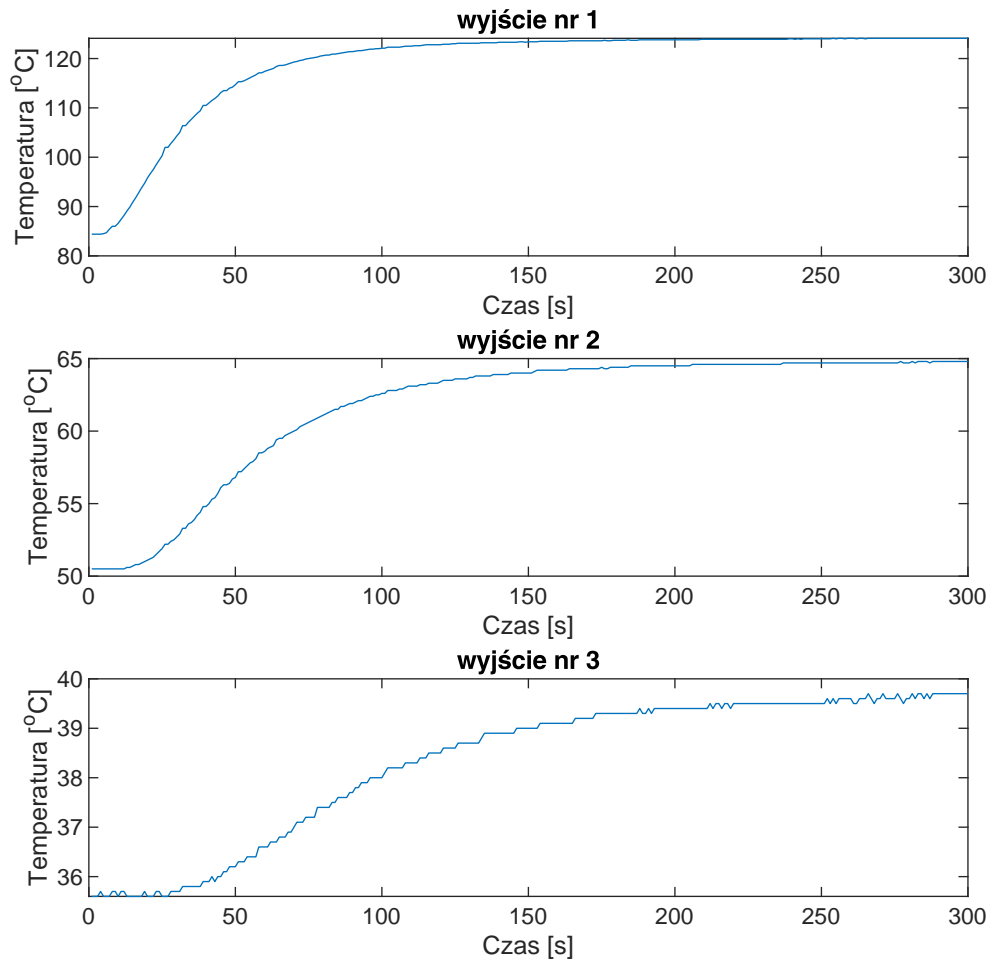
Rys. 2.3. Charakterystyka statyczna eksperymentalnego obiektu cieplnego z jednym wejściem i trzema wyjściami

Rozkład czasowo-przestrzenny odpowiedzi skokowej rozpatrywanego obiektu z jednym wejściem i trzema wyjściami pokazano na rysunku 2.4.



Rys. 2.4. Rozkład czasowo-przestrzenny odpowiedzi skokowej rozpatrywanego obiektu

Większą dokładność odczytu wartości mają wykresy odpowiedzi skokowych pojedynczych wyjść - wykresy pokazane zostały na (Rys. 2.5).



Rys. 2.5. Odpowiedzi skokowe dla wszystkich wyjść obiektu

2.2. Model całkowitego rzędu w przestrzeni stanu

Podstawowym modelem matematycznym całkowitego rzędu opisującym przewodzenie ciepła w obiekcie jednowymiarowym jest cząstkowe równanie różniczkowe parabolicznego typu

z homogenicznymi warunkami brzegowymi Neumanna na końcach, homogenicznymi warunkami początkowymi, wymianą ciepła wzdłuż całej długości pręta oraz rozproszonym sterowaniem oraz obserwacją (zobacz np. [62], [63], [19]):

$$\left\{ \begin{array}{l} \frac{\partial Q(x, t)}{\partial t} = a \frac{\partial^2 Q(x, t)}{\partial x^2} - R_a Q(x, t) + b(x)u(t), \\ \frac{\partial Q(0, t)}{\partial x} = 0, t \geq 0 \\ \frac{\partial Q(1, t)}{\partial x} = 0, t \geq 0 \\ Q(x, 0) = 0, 0 \leq x \leq 1 \\ y(t) = y_0 \int_0^1 Q(x, t)c(x)dx \end{array} \right. \quad (2.1)$$

gdzie $Q(x, t)$ oznacza temperaturę w momencie t i punkcie x , R_a , a oznacza współczynniki przewodzenia ciepła i wymiany ciepła, $b(x)$ oznacza funkcję grzejnika, $c(x)$ jest funkcją czynnika oraz y_0 oznacza stałe wzmocnienie systemu.

Równanie przewodzenia ciepła (2.1) może być przedstawione jako równoważny abstrakcyjny problem początkowy w przestrzeni Hilberta $X = L^2(0, 1)$ ze standardowym iloczynem skalarnym (tzw. model półgrupowy). Podany problem omówił na przykład [64]. W naszym przypadku abstrakcyjna postać równania (2.1) wygląda następująco:

$$\left\{ \begin{array}{l} \dot{Q}(t) = AQ(t) + Bu(t) \\ Q(0) = 0 \\ y(t) = y_0 CQ(t) \end{array} \right. \quad (2.2)$$

gdzie:

$$\left\{ \begin{array}{l} AQ = aQ'' - R_a Q, \\ D(A) = \{Q \in H^2(0, 1) : Q'(0) = 0, Q'(1) = 0\}, \\ a, R_a > 0, \\ H^2(0, 1) = \{u \in L^2(0, 1) : u', u'' \in L^2(0, 1)\}, \\ CQ(t) = \langle c, Q(t) \rangle, Bu(t) = bu(t), \end{array} \right. \quad (2.3)$$

gdzie $\langle u, v \rangle = \int_0^1 u(x)v(x)dx$ oznacza standardowy iloczyn skalarny.

Zbiór wektorów własnych dla operatora stanu A tworzy ortonormalną bazę przestrzeni stanu:

$$h_i(x) = \begin{cases} 0, & i = 0 \\ \sqrt{2} \cos(i\pi x), & i = 1, 2, \dots \end{cases} \quad (2.4)$$

Dyskretne spektrum operatora stanu dla modelu całkowitego rzędu A jest zbiorem pojedynczych, rzeczywistych wartości własnych:

$$\lambda_i = -a\pi^2 i^2 - R_a, i = 0, 1, 2, \dots \quad (2.5)$$

W bazie przestrzeni stanu zdefiniowanej przez zbiór wektorów własnych (2.4) operatory A , B i C mają następującą reprezentację macierzową:

$$A = \text{diag} \{ \lambda_0, \lambda_1, \lambda_2, \dots \} \quad (2.6)$$

$$B = [b_0, b_1, b_2, \dots]^T \quad (2.7)$$

gdzie $b_i = \langle b, h_i \rangle$, $b(x)$ oznacza funkcję elementu grzewczego:

$$b(x) = \begin{cases} 1, & x \in [0, x_0] \\ 0, & x \notin [0, x_0] \end{cases} \quad (2.8)$$

Operator wyjścia ma następującą postać:

$$C = \begin{bmatrix} C_{s1} \\ C_{s2} \\ C_{s3} \end{bmatrix} \quad (2.9)$$

Wiersze operatora wyjścia C są dane następującą formułą:

$$C_{sj} = [c_{sj,0}, c_{sj,1}, c_{sj,2}, \dots] \quad j = 1, 2, 3 \quad (2.10)$$

gdzie $c_{sj,i} = \langle c, h_i \rangle$, $c(x)$ oznacza funkcję czujników wyjściowych:

$$c(x) = \begin{cases} 1, & x \in [x_1, x_2] \\ 0, & x \notin [x_1, x_2] \end{cases} \quad (2.11)$$

Współrzędne x_1 i x_2 zależą od lokalizacji czujnika na pręcie i w rozważanym przypadku są równe:

$$\begin{cases} x = 0.3 : & x_1 = \frac{15}{52}, x_2 = \frac{17}{52} \\ x = 0.5 : & x_1 = \frac{25}{52}, x_2 = \frac{27}{52} \\ x = 0.7 : & x_1 = \frac{36}{52}, x_2 = \frac{38}{52} \end{cases}$$

Ze wzorów (2.8) oraz (2.11) wynika, że funkcja elementu grzewczego $b(x)$ oraz funkcja czujnika wyjściowego $c(x)$ są funkcjami przedziałami stałymi. Załóżmy, że funkcja sterująca

$u(t) = 1(t)$. Wtedy rozwiązanie równania (2.2) posiada postać podobną do rozwiązania obiektu jednowyjściowego przedstawionego na przykład w [62]:

$$y_{IO_j}(t) = y_{0_j} \sum_{i=1}^{\infty} \left(\frac{e^{\lambda_i t} - 1}{\lambda_i} \right) \langle b, h_i \rangle \langle c, h_i \rangle, \quad (2.12)$$

$$j = 1, 2, 3$$

Podstawowe własności omawianego systemu całkowitego rzędu są znane. Można udowodnić (patrz [65]), że operator stanu A dla rozważanego systemu jest: ujemny, samosprężony i ma zwarty operator odwrotny. Można tu również zastosować półgrupy Fellera, które zostały przedstawione w [66], [67] i [15]. Dla znanych współczynników równania a i R_a (2.2) - (2.11) dają dobry opis rozpatrywanego rzeczywistego eksperymentalnego obiektu cieplnego. Przez obcięcie kolejnych elementów operatorów nieskończenie wymiarowych A , B i C uzyskujemy przybliżenie w postaci macierzy skończonego wymiaru, które jest użyteczne do numerycznego modelowania omawianego obiektu. W tym przypadku operatory A , B i C mogą być interpretowane jako macierze. Jeśli te parametry nie są dokładnie znane, można zastosować model przedziałowy (patrz [62], [68], [63]). Dobra dokładność modelu opisanego przez (2.4) - (2.11) zapewnia wielkość przestrzeni stanu równa 25 (patrz [69]), gdzie parametry tego modelu można obliczać za pomocą metody najmniejszych kwadratów. Dla rozpatrywanego obiektu cieplnego wartości macierzy A , B oraz C dla wyjścia nr 2 podane są w tabeli 2.1 (zobacz [19]).

Omówione tu podejście wykorzystujące teorię półgroupową można rozszerzyć na systemy niecałkowitego rzędu. Ten problem został omówiony na przykład przez [15], [70] oraz będzie również rozważany w przypadku systemów niecałkowitego rzędu opisywanych w następnym rozdziale.

2.3. Modele zastępcze całkowitego rzędu w postaci transmitancji

Alternatywnym podejściem do modelowania rozpatrywanego obiektu cieplnego jest zastosowanie modelu transmitacyjnego, przy czym transmitancja jest rozumiana jako:

$$G_j(s) = \frac{Y_j(s)}{U(s)} \quad (2.13)$$

gdzie $G_j(s)$ - transformata Laplace'a j -tego wyjścia, $U(s)$ - transformata Laplace'a sterowania oraz założenie, że transmitancja ma postać jednego ze znanych modeli zastępczych w postaci transmitancji bez opóźnień lub z opóźnieniem, zaproponowanych przez Kupfmuellera lub

Tabela 2.1. Parametry modelu w przestrzeni stanu

Parametry	wartości
rzęd modelu	25
macierz stanu A	$diag\{-0.0271, -0.0364, -0.0644, -0.1110, -0.1763, -0.2603, -0.3629, -0.4841, -0.6240, -0.7826, -0.9598, -1.1556, -1.3702, -1.6033, -1.8551, -2.1256, -2.4148, -2.7225, -3.0490, -3.3941, -3.7578, -4.1402, -4.5413, -4.9610, -5.3993\}$
macierz sterowania B^T	[0.0769 0.1077 0.1046 0.0995 0.0926 0.0842 0.0745 0.0638 0.0526 0.0412 0.0299 0.0190 0.0090 0.0000 -0.0077 -0.0139 - 0.0187 -0.0218 -0.0234 -0.0235 -0.0223 -0.0200 -0.0168 -0.0130 -0.0087]
macierz wyjściowa C_2	[0.9920 0.0000 -1.3995 0.0000 1.3893 0.0000 -1.3724 0.0000 1.3489 0.0000 -1.3191 0.0000 1.2832 0.0000 -1.2415 0.0000 1.1944 0.0000 -1.1423 0.0000 1.0856 0.0000 -1.0248 0.0000 0.9605]

Strejca. Model zastępczy Kupfmuellera opisany jest przez transmitancje:

$$G_1(s) = \frac{ke^{-\tau_1 s}}{T_1 s + 1}$$

$$G_2(s) = \frac{ke^{-\tau_2 s}}{(T_{21} s + 1)(T_{22} s + 1)}$$
(2.14)

W (2.14) k to wzmocnienie statyczne obiektu, τ_1 oraz τ_2 oznacza czas martwy statyczny (czas opóźnienia), T_1 , T_{21} oraz T_{22} są stałymi czasowymi.

Modelowanie transmitancji elementu opóźniającego na platformie MATLAB wymaga od nas zastosowania aproksymacji Pade'go. Rząd P tego przybliżenia musi być również brany

pod uwagę podczas analizy złożoności modelu. Model Strejca transmitancji bez opóźnienia jest opisany jako:

$$G_s(s) = \frac{k}{(T_m s + 1)^M} \quad (2.15)$$

W (2.15) k jest współczynnikiem w stanie ustalonym, T_m oznacza stałą czasową oraz M oznacza rząd aproksymacji.

Parametry obu powyższych transmitancji dla wyjścia nr 2 obiektu podano w tabeli 2.2 (zobacz [19]).

Tabela 2.2. Parametry modeli transmitacyjnych całkowitego rzędu

Parametry	wartości
k	1.1829
$T_1[s]$	38.1392
$T_2[s]$	14.8068
$\tau[s]$	15.0153
rząd aproksymacji Pade P	10
T_m	16.3623
rząd aproksymacji Strejca M	3

3. Modele niecałkowitego rzędu dla rozważanego obiektu

W tym rozdziale rozważymy nowe modele niecałkowitego rzędu opisujące proces rozchodzenia się ciepła w ośrodku jednowymiarowym. W przypadku rozważanego w niniejszej pracy układu doświadczalnego podstawowym modelem matematycznym opisującym to zjawisko jest równanie różniczkowe cząstkowe typu parabolicznego z jednorodnymi warunkami brzegowymi Neumanna na końcach, jednorodnym warunkiem początkowym, wymianą ciepła wzdłuż długości pręta oraz rozproszonym sterowaniem i obserwacją. Równanie z całkowitymi rzędami obu pochodnych jest opisane w rozdziale 2.2.

Wykorzystanie pochodnych niecałkowitego rzędu w badanym modelu można uzasadniać faktem, iż dynamika wymiany ciepła pomiędzy grzejnikiem i prętem oraz pomiędzy prętem a czujnikiem nie jest dokładnie opisana za pomocą równania całkowitego rzędu (2.1). Procesy cieplne zachodzące w obiekcie, np. wymiana ciepła pomiędzy prętem i czujnikiem w obecności rozpraszania ciepła są dość złożone i trudne do dokładnego opisu. Modelowanie matematyczne takich układów i procesów stanowi duże wyzwanie. Rachunek operatorów różniczkowych i całkowych niecałkowitego rzędu, to nieklasyczne podejście matematyczne, które daje różnorodne możliwości ilustracji złożoności zjawisk i procesów występujących w przyrodzie.

3.1. Ciągłe modele niecałkowitego rzędu w przestrzeni stanu

Rozpocniemy od propozycji oraz podania własności nowych ciągłych modeli niecałkowitego rzędu dla procesu przewodnictwa cieplnego w przestrzeni stanu

${}^C D_t^\alpha Q(x, t) = a \frac{\partial^\beta Q(x, t)}{\partial x^\beta} - R_a Q(x, t) + b(x)u(t)$ - niecałkowity rząd pochodnej “po czasie” oraz “po długości”.

Następnie rozważymy dwa uproszczone modele niecałkowitego rzędu:

${}^C D_t^\alpha Q(x, t) = a \frac{\partial^2 Q(x, t)}{\partial x^2} - R_a Q(x, t) + b(x)u(t)$ - niecałkowity rząd pochodnej tylko “po czasie”,

$\frac{\partial Q(x, t)}{\partial t} = a \frac{\partial^\beta Q(x, t)}{\partial x^\beta} - R_a Q(x, t) + b(x)u(t)$ - niecałkowity rząd pochodnej tylko “po długości”.

3.1.1. Ciągły model ułamkowego rzędu względem czasu oraz długości w przestrzeni stanu

Pierwszym rozważanym modelem naszego obiektu cieplnego jest ciągły model niecałkowitego rzędu “po czasie” oraz “po długości”. Przetawiana reprezentacja modelu rzędu niecałkowitego dla procesu cieplnego w przestrzeni stanu dla parametrów rozproszonych została po raz pierwszy opublikowana w pracy [20].

Proponowany model uzyskuje się poprzez zastąpienie pochodnej rzędu pierwszego dla czasu oraz pochodnej drugiego rzędu dla długości odpowiednimi pochodnymi rzędu niecałkowitego. Jest to uzasadnione faktem, iż dynamika przestrzennego rozkładu ciepła wzdłuż grzałki i pręta oraz wzdłuż pręta i czujnika nie jest dokładnie opisana za pomocą równania (2.1). Uważa się, że zastosowanie rzędu niecałkowitego pochodnych lepiej opisuje procesy przewodnictwa cieplnego. Analogiczny sposób zastępowania pochodnych rzędu całkowitego pochodnymi rzędu ułamkowego opisywano w: [1], [2] oraz [3].

Założmy, że pochodna niecałkowitego rzędu w odniesieniu do czasu jest opisana przez definicję Caputo (1.11), a niecałkowita pochodna rzędu w odniesieniu do długości jest opisana przez definicję Riesz (1.23). Następnie równanie wymiany ciepła przechodzi do następującej postaci:

$$\begin{cases} {}^C D_t^\alpha Q(x, t) = a \frac{\partial^\beta Q(x, t)}{\partial x^\beta} - R_a Q(x, t) + b(x)u(t) \\ \frac{\partial Q(0, t)}{\partial x} = 0, t \geq 0 \\ \frac{\partial Q(1, t)}{\partial x} = 0, t \geq 0 \\ Q(x, 0) = 0, 0 \leq x \leq 1 \\ y(t) = y_0 \int_0^1 Q(x, t)c(x)dx \end{cases} \quad (3.1)$$

gdzie $\alpha, \beta > 0$ oznaczają niecałkowity rząd systemu, pozostałe parametry są takie same jak w modelu całkowitego rzędu (2.2). Abstrakcyjna interpretacja systemu (3.1) została przedstawiona w [15]. Proponowane równanie stanu (3.1) można zapisać następująco:

$$\begin{cases} {}^C D_t^\alpha Q(t) = A Q(t) + B u(t) \\ Q(0) = 0 \\ y(t) = y_0 C Q(t) \end{cases} \quad (3.2)$$

gdzie:

$$\begin{cases} AQ = a \frac{\partial^\beta Q(x)}{\partial x^\beta} - R_a Q, \\ D(A) = \{Q \in H^2(0, 1) : Q'(0) = 0, Q'(1) = 0\}, \\ a, R_a > 0, \\ H^2(0, 1) = \{u \in L^2(0, 1) : u', u'' \in L^2(0, 1)\}, \\ CQ(t) = \langle c, Q(t) \rangle, Bu(t) = bu(t) \end{cases} \quad (3.3)$$

Zbiór wektorów własnych dla operatora stanu A tworzy ortonormalną bazę przestrzeni stanu:

$$h_i(x) = \begin{cases} 0, i = 0 \\ \sqrt{2} \cos(i\pi x), i = 1, 2, \dots \end{cases} \quad (3.4)$$

Wartości własne operatora stanu są wyrażone następująco:

$$\lambda_i = -a\pi^\beta i^\beta - R_a, i = 0, 1, 2, \dots \quad (3.5)$$

Operator stanu A ma postać:

$$A = \text{diag} \{\lambda_{\beta_1}, \lambda_{\beta_2}, \lambda_{\beta_3}, \dots\} \quad (3.6)$$

Widmo operatora stanu A jest zatem postaci:

$$\sigma(A) = \{\lambda_{\beta_1}, \lambda_{\beta_2}, \lambda_{\beta_3}, \dots\} \quad (3.7)$$

Zdefiniowany operator A ma dyskretne widmo składające się z rzeczywistych i ujemnych wartości własnych λ_{β_i} , które są powiązane z ortonormalnymi wektorami własnymi (3.4) tworzącymi bazę przestrzeni $L^2(0, 1)$.

Dekompozycja widmowa układu (3.3) jest następująca:

$$\begin{aligned} {}^C D_t^\alpha Q(t) &= AQ(t) + Bu(t) \\ Ah_i &= \lambda_i h_i \\ \langle h_i, h_k \rangle &= \begin{cases} 1, i = k \\ 0, i \neq k \end{cases} \\ {}^C D_t^\alpha Q &= a \frac{\partial^\beta Q}{\partial x^\beta} - R_a Q + Bu \\ {}^C D_t^\alpha \sum_{i=0}^{\infty} c_i h_i &= a \frac{\partial^\beta \sum_{i=0}^{\infty} c_i h_i}{\partial x^\beta} - R_a \sum_{i=0}^{\infty} c_i h_i + Bu/h_i \end{aligned}$$

$${}^C D_t^\alpha c_i = a \frac{\partial^\beta c_i}{\partial x^\beta} - R_a c_i + b_i u \quad (3.8)$$

W bazie przestrzeni stanu zdefiniowanej przez zbiór wektorów własnych (3.4) operator B ma następującą reprezentację macierzową:

$$B = [b_0, b_1, b_2, \dots]^T \quad (3.9)$$

gdzie $b_i = \langle b, h_i \rangle$, $b(x)$ oznacza funkcję elementu grzewczego:

$$b(x) = \begin{cases} 1, & x \in [0, x_0] \\ 0, & x \notin [0, x_0] \end{cases} \quad (3.10)$$

Operator wyjścia ma następującą postać:

$$C = \begin{bmatrix} C_{s1} \\ C_{s2} \\ C_{s3} \end{bmatrix} \quad (3.11)$$

Wiersze operatora wyjścia C są dane następującą formułą:

$$C_{sj} = [c_{sj,0}, c_{sj,1}, c_{sj,2}, \dots] \quad j = 1, 2, 3 \quad (3.12)$$

gdzie $c_{sj,i} = \langle c, h_i \rangle$, $c(x)$ oznacza funkcję czujników wyjściowych:

$$c(x) = \begin{cases} 1, & x \in [x_1, x_2] \\ 0, & x \notin [x_1, x_2] \end{cases} \quad (3.13)$$

Ze wzorów (3.10) oraz (3.13) wynika, że funkcja elementu grzewczego $b(x)$ oraz funkcja czujnika wyjściowego $c(x)$ są funkcjami przedziałami stałymi. Współrzędne x_1 i x_2 zależą od lokalizacji czujnika na przecie i w rozważanym przypadku są podane w (2.2).

Teraz możemy zastosować teorię półgrup Fellera, które zostały przedstawione w [15], [66] oraz [71].

Półgrupa zdefiniowana jest następująco:

Definicja 20. Półgrupa

Półgrupa jest to zbiór S wraz z operacją binarną $T(T : S \times S \rightarrow S)$, która jest łączna, tzn. $\forall x, y, z \in S, \quad T(T(x, y), z) = T(x, T(y, z))$.

Unikalność rozwiązania dowodzi własność półgrupy, która jest podana przez następującą formułą:

$$T(t + s) = T(t)T(s), \quad t, s > 0 \quad (3.14)$$

Własność półgrupy podana w zależności (3.14) dla rodziny funkcji, $\{T(t) : t \geq 0\}$, jest złożeniem. Zauważmy, że $T(0)$ jest operatorem tożsamości (Id). Silnie ciągła dodatnia kontrakcja półgrup na $C_\infty(S)$ jest nazywana półgrupą Fellera na S . Rozważamy reprezentację operatorów różniczkowych, które tworzą zbiór półgrup Fellera. Z rezultatów prezentowanych przez Popescu w [15] możemy wywnioskować, że system niecałkowitego rzędu jest rozwiązaniem problemu Cauchy'ego

$${}^C D_t^\alpha Q(t) = A Q(t) + B u(t), \quad Q(0) = 0$$

gdzie $0 < \alpha < 1, t \geq 0$ oraz A jest generatorem ograniczonej ciągłej półgrupy Fellera $T(t)_{t \geq 0}$ w przestrzeni Hilberta $H^\beta(0, 1)$.

Należy zauważyć, że przedstawione powyżej podejście wykorzystujące teorię półgrup, które można rozszerzyć na system niecałkowitego rzędu omawiane jest m.in. w [15] oraz [70].

Elementarne własności dla systemów parabolicznych całkowitego rzędu, które zostały przeanalizowane oraz udowodnione to: operator stanu A dla systemu całkowitego rzędu opisanego przez (2.6) jest: ujemny, samosprężony i ma zwarty operator odwrotny (zobacz [65]).

Odpowiednie twierdzenie dla $\beta = 2$ jest oczywiście prawdziwe. Podobne twierdzenie, dla wszystkich β , jest prawdziwe w pewnych warunkach początkowych (patrz [72]).

Załóżmy, że funkcja sterująca $u(t) = 1(t)$. Rozwiązanie równania stanu (3.2) można obliczyć za pomocą transformaty Laplace'a dla operatora Caputo zdefiniowanego przez (1.13) przy założeniu, że warunek początkowy jest równy zeru: $Q(x, 0) = 0, 0 \leq x \leq 1$ oraz operatory stanu i sterowania są opisane przez (3.6) - (3.10).

Jeżeli przyjmiemy, że sygnał sterujący ma postać funkcji Heaviside'a $u(t) = 1(t)$ i zastosujemy (1.14), to otrzymamy rozwiązanie (3.2) w następującej postaci:

$$y_j(t) = y_{0j} \sum_{i=1}^{\infty} \frac{(E_\alpha(\lambda_{\beta_i} t^\alpha) - 1(t))}{\lambda_{\beta_i}} \langle b, h_i \rangle \langle c, h_i \rangle, \quad (3.15)$$

$$j = 1, 2, 3$$

i w konsekwencji macierz wyjścia ma postać:

$$y(t) = [y_1(t), y_2(t), y_3(t)]^T. \quad (3.16)$$

Model niecałkowitego rzędu podany wyżej przez (3.2) - (3.15) jest modelem nieskończenie wymiarowym. Jego praktyczne zastosowanie wymaga od nas użycia skończenie wymiarowej aproksymacji. Można to uzyskać pomijając dalsze elementy w równaniu stanu (3.2) i w konsekwencji obliczając rozwiązanie (3.15) i (3.16) jako skończoną sumę wyrażoną przez (3.17). W konsekwencji operatory A, B i C są interpretowane jako macierze.

$$y_j(t) = y_{0_j} \sum_{i=1}^N \frac{(E_\alpha(\lambda_{\beta_i} t^\alpha) - 1(t))}{\lambda_{\beta_i}} \langle b, h_i \rangle \langle c, h_i \rangle, \quad (3.17)$$

$$j = 1, 2, 3$$

W (3.17) N oznacza wymiar aproksymacji. W rzeczywistych warunkach jego wartość niezbędną dla uzyskania założonej dokładności można oszacować za pomocą symulacji. Jej prawidłowe oszacowanie jest kluczowym problemem podczas korzystania z przedstawionych modeli i zostanie omówione w kolejnych rozdziałach.

Badania przeprowadzone z wykorzystaniem doświadczalnego układu cieplnego pokazanego na rysunku 2.1. Testowano odpowiedź skokową modelu w przedziale czasowym od 0 do 300 s z czasem próbkowania 1 s. Wartości parametrów obliczono optymalizując funkcję MSE (Medium Square Error) wg (3.18) - dla każdego badanego wymiaru N oddzielnie.

$$I_{MSE} = \sum_{j=1}^3 \int_0^{T_f} (y_j(t) - y_{e_j}(t))^2 dt \quad (3.18)$$

gdzie $y_{e_j}(t)$ - dane rzeczywistego obiektu.

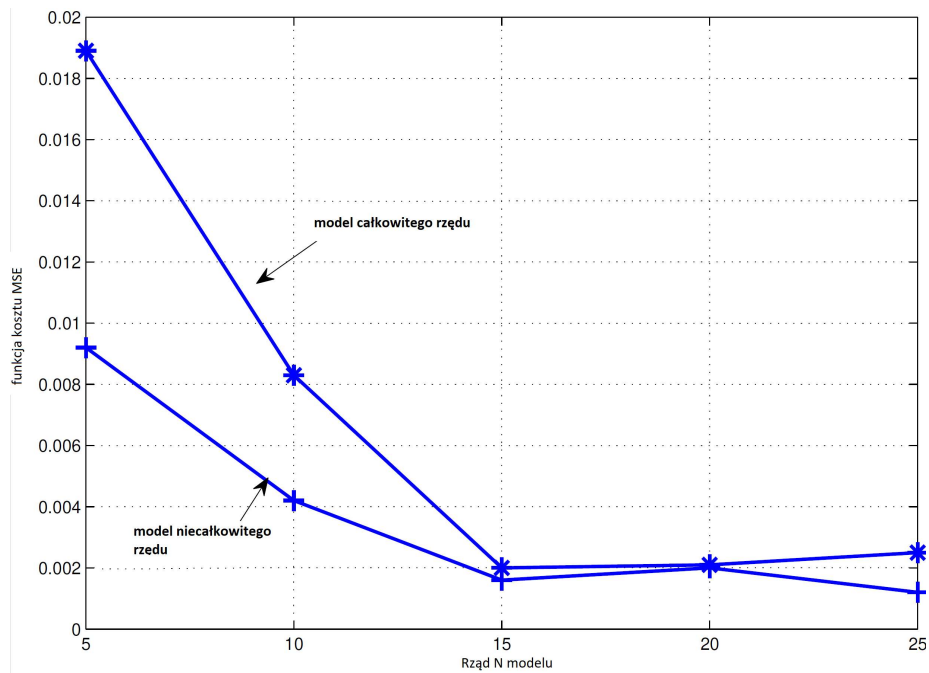
Zaproponowany model porównano z modelem całkowitego rzędu wyrażonym wzorami (2.1) - (2.12). Wyniki podano w tabeli 3.1.

Tabela 3.1. Funkcja kosztu (3.18) dla modelu całkowitego oraz niecałkowitego rzędu oraz różnych wartości N

N	Model całkowitego rzędu	Model niecałkowitego rzędu
5	0.0189	0.0092
10	0.0083	0.0042
15	0.0020	0.0016
20	0.0021	0.0020
25	0.0025	0.0012

Funkcję kosztu (3.32) jako funkcję N dla obu omawianych modeli przedstawiono na rysunku 3.1, optymalne parametry modelu rzędu liczb całkowitych podano w tabeli 3.2, optymalne parametry dla modelu rzędu niecałkowitego podano w tabeli 3.3.

Błąd modelu w funkcji czasu i długości pręta przedstawiono na rysunkach 3.4 i 3.5. Rozkład czasowo-przestrzenny temperatury obliczony z wykorzystaniem obu rozpatrywanych modeli pokazano na rysunkach 3.2 i 3.3.



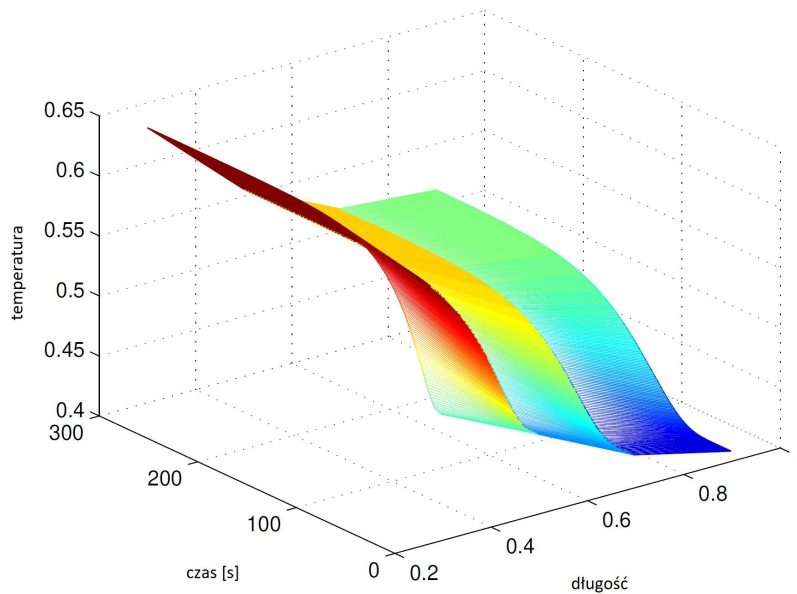
Rys. 3.1. Funkcja kosztu (3.18) jako funkcja wartości rzędu N

Tabela 3.2. Optymalne wartości parametrów modelu całkowitego rzędu dla różnych N

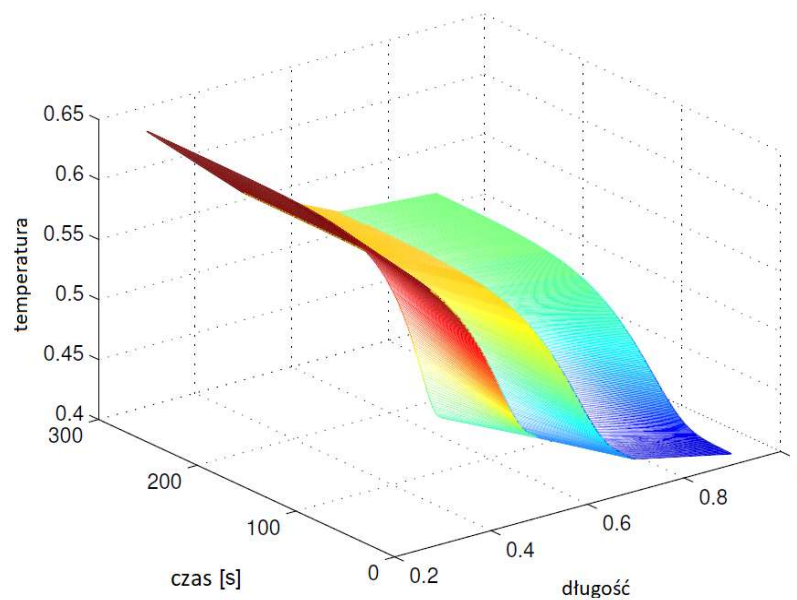
N	a_w	R_a
5	0.0017	0.0224
10	0.0012	0.0253
15	0.0009	0.0317
20	0.0009	0.0312
25	0.0009	0.0308

Tabela 3.3. Optymalne wartości parametrów modelu niecałkowitego rzędu dla różnych N

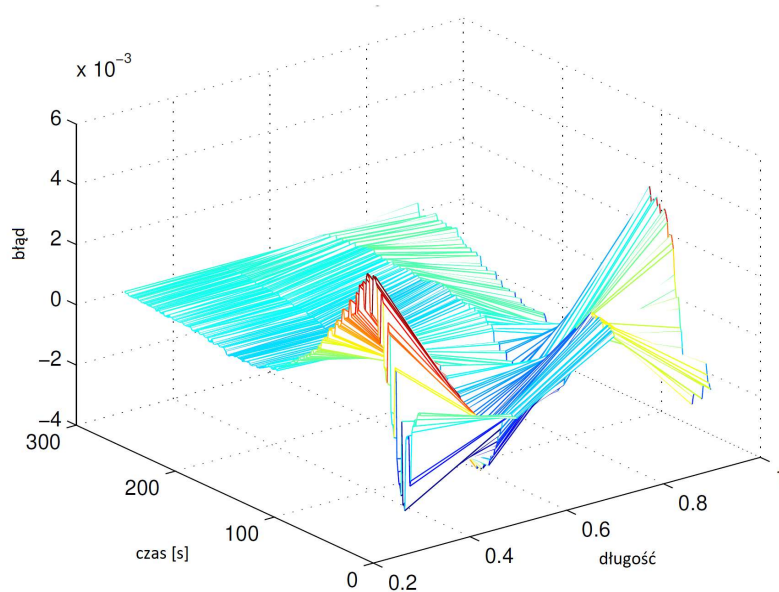
N	α	β	a_w	R_a
5	1.0571	2.1643	0.0012	0.0173
10	1.0328	2.0526	0.0010	0.0224
15	0.9896	1.9930	0.0009	0.0343
20	0.9982	2.0018	0.0009	0.0316
25	0.9744	1.9864	0.0007	0.0423



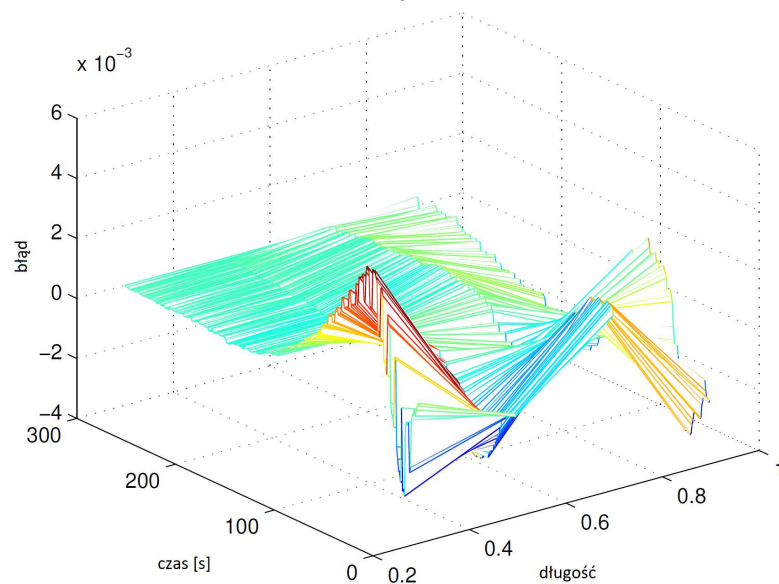
Rys. 3.2. Czasowo-przestrzenny rozkład temperatury dla modelu całkowitego rzędu i $N = 15$



Rys. 3.3. Czasowo-przestrzenny rozkład temperatury dla modelu niecałkowitego rzędu i $N = 15$



Rys. 3.4. Błąd modelu całkowitego rzędu w funkcji czasu i długości pręta dla $N = 15$



Rys. 3.5. Błąd modelu niecałkowitego rzędu w funkcji czasu i długości pręta dla $N = 15$

Z tabeli 3.1 oraz rysunku 3.1 można wywnioskować, że proponowany ciągły model rzędu niecałkowitego jest dokładniejszy w sensie funkcji kosztu MSE (3.18) dla każdego z rozważanych rzędów aproksymacji N . Dobrą dokładność model osiąga dla rzędu aproksymacji $N = 15$, co daje rozsądny rząd modelu podczas jego stosowania.

Można stwierdzić, że proponowany model niecałkowitego rzędu w przestrzeni stanów dla jednowymiarowego obiektu cieplnego można zbudować poprzez uogólnienie całkowitego, abstrakcyjnego modelu w przestrzeni Hilberta z wykorzystaniem półgrup Fellera. Co więcej, dokładność proponowanego modelu w sensie funkcji kosztu MSE jest lepsza niż analogowego modelu rzędu całkowitego dla każdego wymiaru jego skończonej aproksymacji. Parametry proponowanego modelu można obliczyć numerycznie na podstawie wyników eksperymentów.

Dalsze badania przedstawionych zagadnień będą dotyczyły uogólnienia wyników na niepewność parametrów oraz formułowania i rozwiązywania warunków dekompozycji widma dla rozpatrywanej klasy nieskończenie wymiarowych modeli niecałkowitego rzędu.

3.1.2. Uprozczone modele ułamkowego rzędu w przestrzeni stanu

Zauważmy, że proponowany model niecałkowitego rzędu opisany po raz pierwszy w [21] przez równania (3.1) - (3.15) dla rzędów: $\alpha = 1$ i $\beta = 2$ zamienia się w model całkowitego rzędu (2.1) - (2.12).

Podstawienie: $\alpha = 1$ lub $\beta = 2$ pozwala nam natomiast uzyskać odpowiednie „modele częściowego rzędu niecałkowitego” z pochodnymi rzędu niecałkowitego, odpowiednio w odniesieniu do długości lub czasu.

Na początku rozważmy ciągły model ułamkowego rzędu względem czasu w przestrzeni stanu.

Załóżmy, że operator niecałkowitego rzędu jest opisany przez definicję Caputo (1.11). Wtedy równanie wymiany ciepła posiada następującą postać:

$$\left\{ \begin{array}{l} {}^C D_t^\alpha Q(x, t) = a \frac{\partial^2 Q(x, t)}{\partial x^2} - R_a Q(x, t) + b(x)u(t) \\ \frac{\partial Q(0, t)}{\partial x} = 0, t \geq 0 \\ \frac{\partial Q(1, t)}{\partial x} = 0, t \geq 0 \\ Q(x, 0) = 0, 0 \leq x \leq 1 \\ y(t) = y_0 \int_0^1 Q(x, t)c(x)dx \end{array} \right. \quad (3.19)$$

gdzie $0 < \alpha < 2$ oznacza niecałkowity rząd obiektu, pozostałe parametry są takie same jak w modelu całkowitego rzędu podanym w (2.2).

Równanie (3.19) można przedstawić jako równanie stanu w przestrzeni Hilberta analogicznie do (2.2). Będzie ono miało postać:

$$\begin{cases} {}^C D_t^\alpha Q(t) = AQ(t) + Bu(t) \\ Q(0) = 0 \\ y(t) = y_0 CQ(t) \end{cases} \quad (3.20)$$

gdzie:

$$\begin{cases} AQ = aQ'' - R_a Q, \\ D(A) = \{Q \in H^2(0, 1) : Q'(0) = 0, Q'(1) = 0\}, \\ a, R_a > 0, \\ H^2(0, 1) = \{u \in L^2(0, 1) : u', u'' \in L^2(0, 1)\}, \\ CQ(t) = \langle c, Q(t) \rangle, Bu(t) = bu(t) \end{cases} \quad (3.21)$$

Elementarne własności dla systemów parabolicznych całkowego rzędu, które zostały przeanalizowane oraz udowodnione to następujące fakty: operator stanu A dla systemu całkowego rzędu opisanego przez (2.6) jest: ujemny, samosprężony i ma zwarty operator odwrotny (zobacz [73]).

Teraz zostanie udowodniony fakt, iż analogiczne właściwości są prawdziwe także dla systemu niecałkowitego rzędu.

Propozycja 1. *Podstawowe własności operatora stanu A*

Operator stanu A dla systemu niecałkowitego rzędu opisany przez równanie (3.20) jest: ujemny, samosprężony i ma zwarty operator odwrotny.

Dowód. Weźmy $P \in H^2(0, 1)$. Zgodnie z (3.21) mamy

$$\begin{aligned} \langle P, AQ \rangle &= \int_0^1 P(x)[aQ''(x) - R_a Q(x)]dx = \\ &= a \int_0^1 P(x)Q''(x)dx - R_a \langle P, Q \rangle = \\ &= a(P(1)Q'(1) - P(0)Q'(0)) \\ &= -a \int_0^1 P'(x)Q'(x)dx - R_a \langle P, Q \rangle = \\ &= -aQ(1)P'(1) \\ &+ \int_0^1 [aP''(x) - R_a P(x)]Q(x)dx \\ &\supset Q \in D(a) \end{aligned} \quad (3.22)$$

Korzystając z twierdzenia Riesz'a o reprezentacji funkcjonału liniowego i ograniczonego w przestrzeni Hilberta ([74]) funkcjonał liniowy

$$H^2(0,1) \supset D(A) \ni Q \rightarrow \int_0^1 [aP''(x) - R_a P(x)Q(x)]dx \in H^2(0,1) \quad (3.23)$$

przedłuża się do funkcjonału liniowego i ograniczonego na $H^2(0,1)$.

Pozostałe wyrażenia reprezentują funkcjonał liniowy, gęsto zdefiniowany na $H^2(0,1)$, przedłużalny do funkcjonału liniowego i ograniczonego określonego na całym H , tylko jeśli $p'(0) = 0$ oraz $p'(1) = 0$. Zatem zgodnie z definicją operatora sprzężonego [74] mamy

$$D(A^*) = \{P \in H^2(0,1) : P'(0) = 0, P'(1) = 0\} = D(A) \quad (3.24)$$

oraz

$$A^*P = aP'' - R_a P = AP \quad (3.25)$$

co oznacza, że operator A jest samosprzężony.

Ponadto

$$\begin{aligned} \langle Q, AQ \rangle &= \int_0^1 Q(x)[aQ''(x) - R_a Q(x)]dx = \\ &= aQ(1)Q'(1) - aQ(0)Q'(0) - a \int_0^1 [Q'(x)]^2 dx - R_a \|Q^2\| \\ &\leq -R_a \|Q^2\| \supset Q \in D(A) \end{aligned} \quad (3.26)$$

zatem operator $-A + R_a I$ jest nieujemny.

Aby udowodnić zwartość A^{-1} , rozważmy operator pomocniczy G :

$$GQ = Q'', D(G) = D(A).$$

Wyznaczenie operatora odwrotnego G^{-1} wymaga rozwiązania układu

$$\begin{cases} Q''(x) = f(x), 0 \leq x \leq 1 \\ Q'(1) = 0 \\ Q'(0) = 0 \end{cases} \quad (3.27)$$

Układ posiada dokładnie jedno rozwiązanie

$$Q(x) = - \int_0^1 yf(y)dy + \int_0^x (x-y)f(y)dy$$

Zatem operator G^{-1} istnieje i ma następującą postać:

$$G^{-1} = \langle g, f \rangle h + Kf,$$

gdzie

$$\begin{aligned} g, h &\in H^2(0, 1), \\ g(x) &= x, \\ h(x) &= -1, 0 \leq x \leq 1, \end{aligned} \quad (3.28)$$

oraz K jest operatorem całkowym

$$(Kf)(x) = \int_0^x k(x-y)f(y)dy$$

z jądrem

$$k(x, y) = \begin{cases} x - y, & 0 \leq y \leq x \\ 0, & x \leq y \leq 1 \end{cases} \quad (3.29)$$

K jest operatorem Hilberta-Schmidta, więc operator G^{-1} jest zwarty.

Stąd G posiada zwartą rezolwentę $(\lambda I - G)^{-1}$, oraz

$$A^{-1} = \frac{1}{a} \left(-\frac{R_a}{a} I + G \right)^{-1}.$$

Ostatecznie operator A^{-1} jest zwarty.

□

Rozwiązanie równania stanu (3.19) można obliczyć za pomocą transformaty Laplace'a dla operatora C zdefiniowanego przez (1.13) z założeniami, że warunek początkowy jest równy 0: $Q(x, 0) = 0, 0 \leq x \leq 1$ oraz operatory stanu i sterowania opisane są przez (2.6) - (2.8). Jeśli przyjmiemy, że sygnał sterujący ma postać funkcji Heaviside $u(t) = 1(t)$ i stosuje się (1.14), wówczas otrzymamy rozwiązanie w następującej formie:

$$y_{FO}(t) = y_0 \sum_{n=1}^{\infty} \frac{(E_{\alpha}(\lambda_n t^{\alpha}) - 1(t))}{\lambda_n} \langle b, h_n \rangle \langle c, h_n \rangle \quad (3.30)$$

Zauważmy, że to samo rozwiązanie można obliczyć za pomocą (1.31):

$$\begin{aligned} Q(t) &= \Phi_0(t)x_0 + \int_0^t \Phi(t-\tau)Bd\tau = \\ &= \sum_{k=0}^{\infty} \frac{A^k t^{k\alpha}}{\Gamma(k\alpha + 1)} x_0 + \int_0^t \sum_{k=0}^{\infty} \frac{A^k (t-\tau)^{(k+1)\alpha-1}}{\Gamma((k+1)\alpha)} Bd\tau = \\ &= \sum_{k=0}^{\infty} \frac{A^k t^{k\alpha}}{\Gamma(k\alpha + 1)} x_0 + B \sum_{k=0}^{\infty} \frac{A^k t^{(k+1)\alpha}}{\Gamma((k+1)\alpha)(k+1)\alpha} \end{aligned} \quad (3.31)$$

Eksperymenty przeprowadzono za pomocą eksperymentalnego systemu pokazanego na rysunku 2.1. Odpowiedź skokowa systemu została przetestowana w zakresie czasu od 0 do $T_f = 300s$, parametry proponowanego nowego modelu zostały obliczone przy użyciu metody najmniejszych kwadratów i typowej funkcji kosztu :

$$I_1 = \int_0^{T_f} (y - y_e)^2 dt \quad (3.32)$$

Parametry optymalne w sensie powyższej funkcji kosztowej podane są w tabeli 3.4, diagramy bezwzględnego błędu aproksymacji dla wielkości modelu $N = 15$ i $N = 25$ podane są na rysunkach 3.6 i 3.7. Porównanie jest również zilustrowane przez wyniki dla modelu rzędu całkowitego podanego w tabeli 3.4 oraz pokazane na rysunku 3.8.

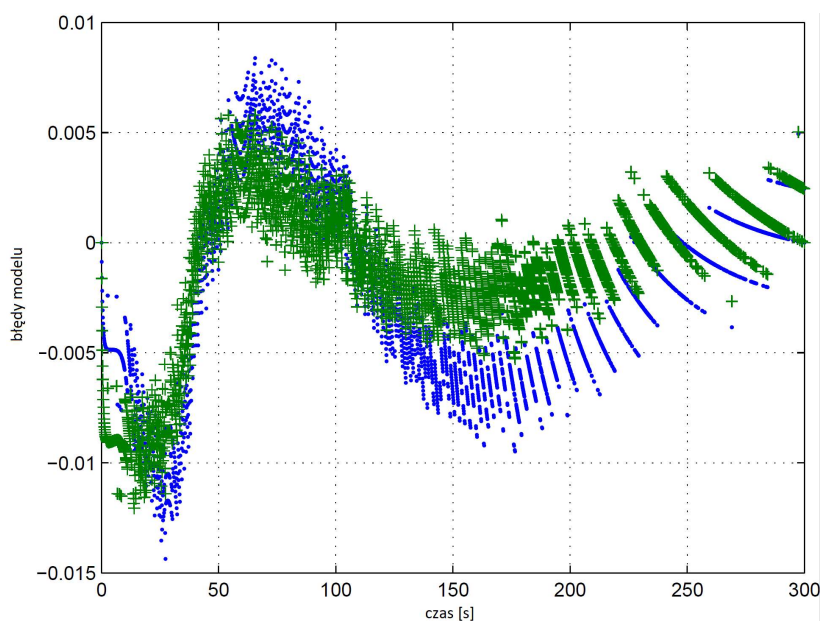
Tabela 3.4. Parametry modeli całkowitego i ułamkowego rzędu oraz wartość funkcji kosztu (3.32).

Model	całkowity rząd		Niecałkowity rząd	
	15	25	15	25
rząd N	15	25	15	25
α	1	1	0.9752	0.9706
a_w	0.0009	0.0009	0.0008	0.0009
R_a	0.0275	0.0274	0.0385	0.0391
Funkcja kosztu(3.32)	0.0072	0.0056	0.0036	7.7362e-04

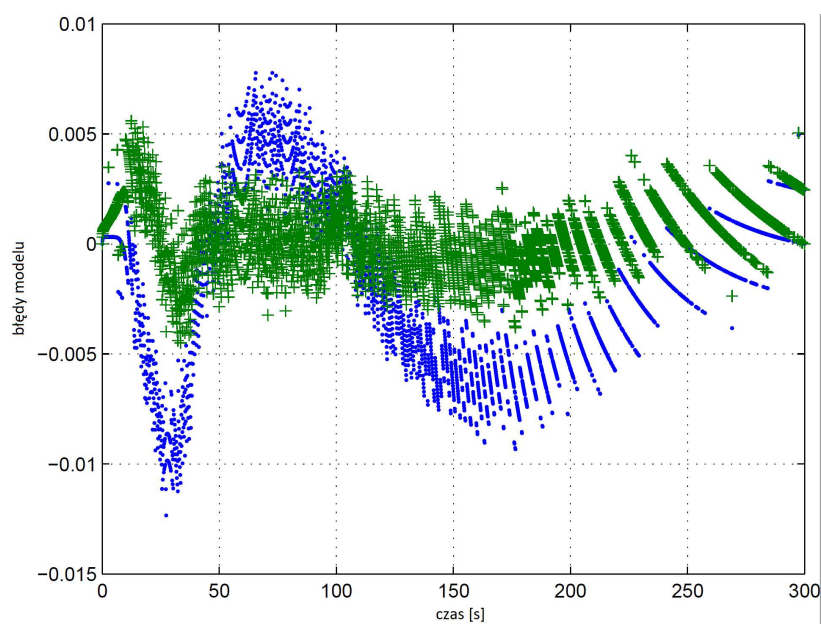
Z tabeli 3.4 oraz rysunków 3.6, 3.7, 3.8 można wywnioskować od razu, że zaproponowany model niecałkowitego rzędu lepiej opisuje rozważany obiekt nieskończenie wymiarowy, niż model całkowitego rzędu przedstawiony wcześniej. Podsumowując można stwierdzić, że zaproponowany model ułamkowego rzędu w przestrzeni stanu dla jednowymiarowego procesu przewodnictwa cieplnego, można zbudować poprzez uogólnienie abstrakcyjnego modelu rzędu całkowitego. Ponadto dokładność proponowanego modelu w sensie funkcji kosztu minimalno-kwadratowego jest lepsza niż analogiczny model rzędu całkowitego. Parametry proponowanego modelu można obliczyć numerycznie.

Dalsza analiza przedstawionego problemu obejmuje propozycję oraz testowanie ciągłego modelu w przestrzeni stanu z niecałkowitą pochodną przestrzenną, opisaną przez operator Riesz - tzw. ciągły model obiektu cieplnego niecałkowitego rzędu "względem długości". Proponowany model niecałkowitego rzędu w odniesieniu do współrzędnych przestrzennych uzyskuje się przez zastąpienie pochodnej przestrzennej drugiego rzędu odpowiednią pochodną rzędu niecałkowitego:

$$\frac{\partial Q(x, t)}{\partial t} = a \frac{\partial^\beta Q(x, t)}{\partial x^\beta} - R_a Q(x, t) + b(x)u(t).$$

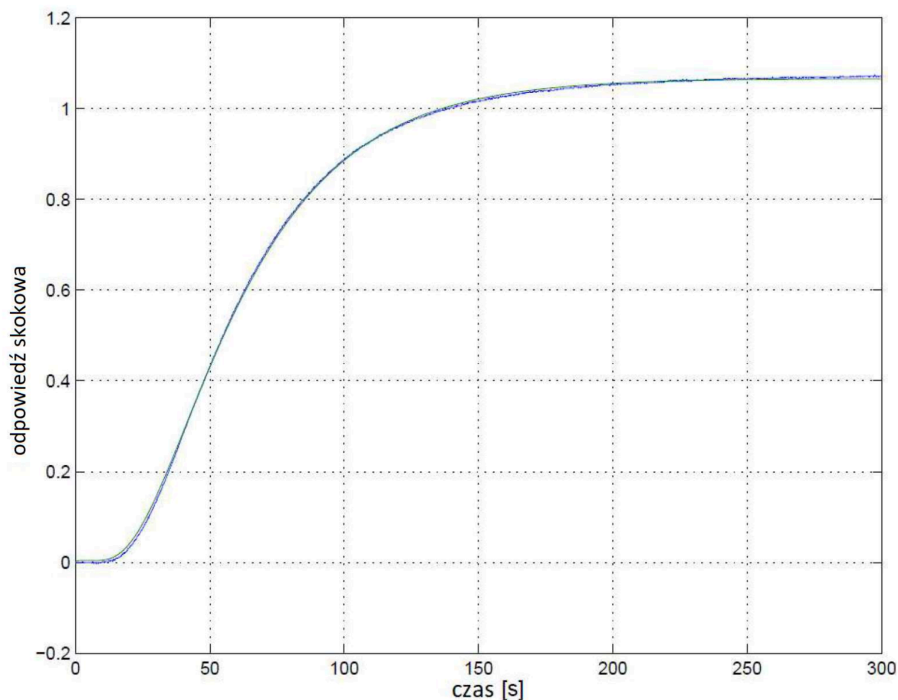


Rys. 3.6. Błąd dla modeli z rzędem całkowitym (.) oraz niecałkowitym (+), rząd $N = 15$.



Rys. 3.7. Błąd dla modeli z rzędem całkowitym (.) oraz niecałkowitym (+), rząd $N = 25$.

Model ten zbadamy eksperymentalnie w następnym rozdziale, podstawiając w modelu niecałkowitego rzędu zdefiniowanym wzorami (3.1) - (3.3) wartość $\alpha = 1$.



Rys. 3.8. Odpowiedź skokowa modelu rzeczywistego obiektu oraz modelu niecałkowitego rzędu dla $N = 25$.

3.1.3. Porównanie oraz identyfikacja wartości parametrów ciągłych modeli w przestrzeni stanu

Problemy identyfikacji modeli były szeroko omawiane przez wielu autorów, np. w fundamentalnej książce [75], przykładowe identyfikacje modeli rzędów ułamkowych podano w [10] i [76]. Klasyczne podejście polega na zminimalizowaniu funkcji kosztu jako funkcji parametrów modelu. Funkcja kosztu opisuje różnicę między odpowiedzią modelu a odpowiedzią rzeczywistego obiektu. Funkcje kosztów określające jakość proponowanego modelu w przestrzeni stanów są wymagane do opisanego zarówno czasowego, jak i przestrzennego rozkładu temperatury w obiekcie. Ponadto funkcję kosztu należy przypisać na siatce czasu dyskretnego, ponieważ zarówno pomiary, jak i obliczenia modelowe są dyskretne.

Założmy, że wyjście systemu $y(t)$ zostało zsynchronizowane z czasem próbkowania T_s , maksymalna liczba próbek czasowych zebranych podczas eksperymentu była równa K_s . Miara w k -tym momencie czasowym i j -tym wyjściu będzie oznaczona jako $y_{p_j}^+(k) = y(kT_s)$, $k = 1, \dots, K_s$, a odpowiedź modelu w tym samym momencie jest oznaczona przez $y_j^+(k)$.

Niech liczba swobodnych parametrów modelu będzie równa M_F . W naszym przypadku liczba ta wynosi 4, ponieważ wszystkie parametry: α , β , a i R_a służą do dopasowania odpowiedzi modelu do odpowiedzi obiektu.

Następnie wyznaczono różnicę między odpowiedzią modelu, a odpowiedzią obiektu rzeczywistego w k -tym momencie czasowym i przy j -tym wyjściu o $e_j^+(k)$:

$$e_j^+(k) = y_{p_j}^+(k) - y_j^+(k) \quad (3.33)$$

Funkcja kosztu MSE w rozważanym wypadku przyjmuje następującą postać ([75]):

$$MSE = \frac{1}{3K_s} \sum_{j=1}^3 \sum_{k=1}^{K_s} e_j^+(k) \quad (3.34)$$

Wszystkie parametry: α , β , a i R_a modelu 3.3 zostały obliczone z pomocą środowiska Matlab przy użyciu funkcji *fminsearch*, która minimalizuje koszt (3.34).

Innym ważnym problemem jest oszacowanie rzędu skończonego wymiaru modelu N . Można to zrobić za pomocą podejścia zaprezentowanego np. w [75]. Polega on na obliczeniu funkcji kosztu (3.34) dla rosnących wartości rzędu N . Jeśli obserwuje się znaczną poprawę funkcji kosztu podczas przejścia z wartości N do $N + 1$, to oszacowany rząd modelu jest równy $N + 1$. Wyniki tak przeprowadzonych testów podano w tabeli 3.5.

Eksperymenty dla wszystkich ciągłych modeli definiowanych w przestrzeni stanu dla rozważanego procesu przewodnictwa cieplnego, zarówno całkowitego jak i niecałkowitego rzędu, prezentowanych w rozdziale 2 oraz 3 niniejszej rozprawy, zostały wykonane przy użyciu systemu laboratoryjnego przedstawionego w rozdziale 2 i pokazanego na rysunku 2.1. Wyniki tych badań zostały opublikowane w [23].

Porównywane modele oznaczmy:

- model 1 - całkowity rząd, $\alpha = 1$, $\beta = 2$
- model 2 - niecałkowity rząd względem długości, $\alpha = 1$, $\beta \in \mathbb{R}$
- model 3 - niecałkowity rząd względem czasu, $\alpha \in \mathbb{R}$, $\beta = 2$
- model 4 - niecałkowity rząd względem czasu i długości, $\alpha \in \mathbb{R}$, $\beta \in \mathbb{R}$

Podczas badań testowano odpowiedź skokową pomiędzy 40% a 60% maksymalnej wartości sygnału sterującego podawanego na obiekt. Rozkład temperatury w czasie i przestrzeni jest pokazany na rysunku 2.4. Podczas testów czas próbkowania był równy $T_s = 1s$, liczba próbek czasu była równa $K_s = 300$.

Proponowane modele niecałkowitego rzędu (model 2, model 3, model 4) zostały porównane z podstawowym modelem całkowitego rzędu (model 1). Wyniki podano w tabeli 3.5.

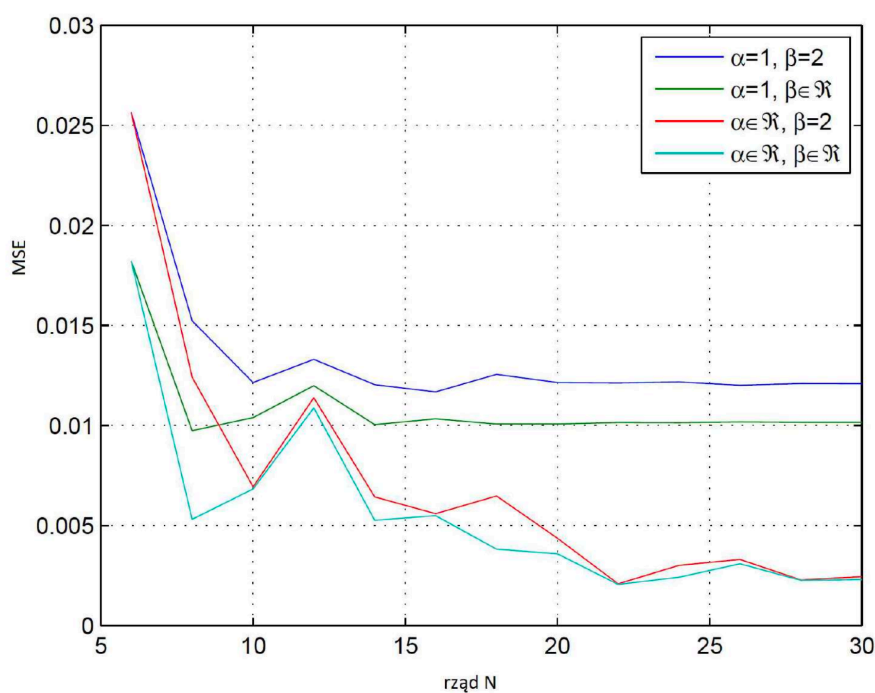
Tabela 3.5. Funkcja kosztu MSE (3.34) dla różnych N oraz wszystkich testowanych modeli

Rząd N	model 1 $\alpha = 1,$ $\beta = 2$	model 2 $\alpha = 1,$ $\beta \in \mathbb{R}$	model 3 $\alpha \in \mathbb{R},$ $\beta = 2$	model 4 $\alpha \in \mathbb{R},$ $\beta \in \mathbb{R}$
6	0.296082	0.210358	0.296048	0.210324
8	0.175856	0.112488	0.143381	0.061347
10	0.140216	0.120017	0.080125	0.078878
12	0.153525	0.138439	0.131474	0.125623
14	0.138942	0.115968	0.074268	0.060765
16	0.134770	0.119325	0.064554	0.063440
18	0.144883	0.116270	0.074788	0.044133
20	0.140243	0.116270	0.050416	0.041390
22	0.139945	0.117205	0.024154	0.007108
24	0.140586	0.117074	0.034738	0.010223
26	0.138594	0.117506	0.038179	0.112364
28	0.139619	0.117264	0.026374	0.007762
30	0.139486	0.117294	0.028256	0.008315

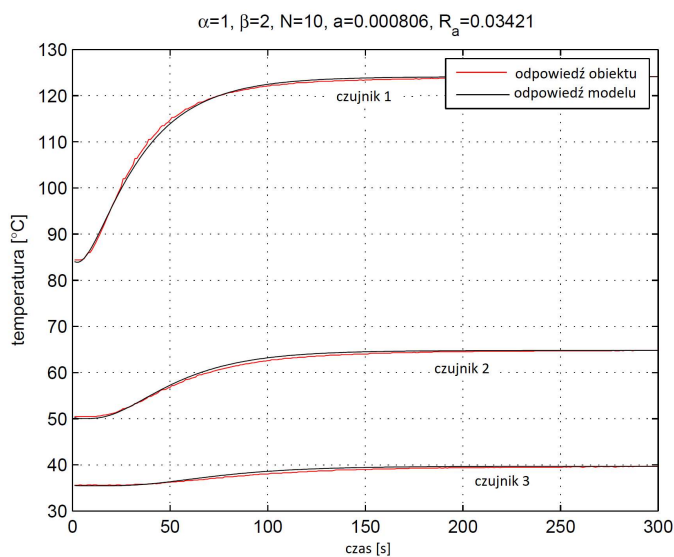
Wartość MSE jako funkcję rzędu N dla wszystkich testowanych modeli pokazano na rysunku 3.9.

Z tabeli 3.5 i rysunku 3.9 można wywnioskować, że najlepszym modelem opisującym nasz obiekt (w sensie minimalizacji funkcji kosztu MSE) jest model z rzędem niecałkowitym ($\alpha, \beta \in \mathbb{R}$).

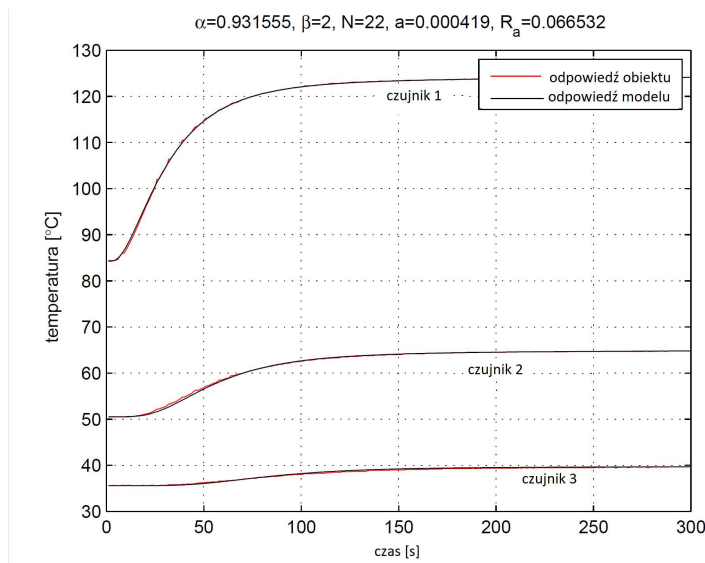
Na rysunkach 3.10-3.13 jest porównana odpowiedź skokowa wszystkich czterech modeli z rzeczywistą odpowiedzią dla wcześniej wybranego optymalnego rzędu N .



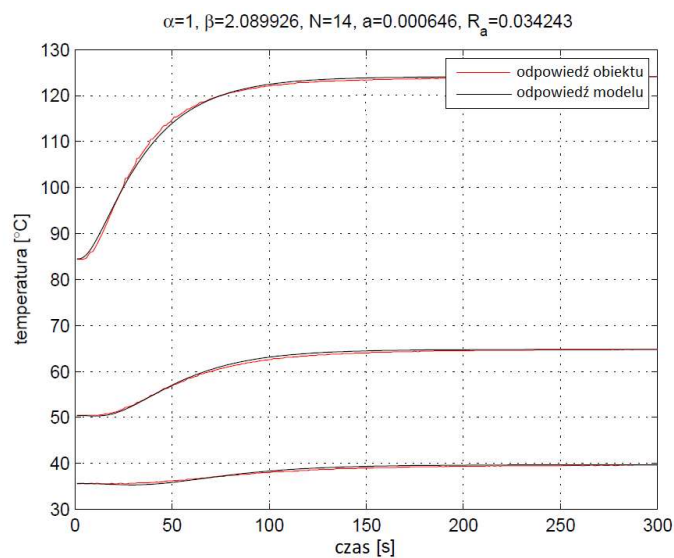
Rys. 3.9. Zależność wartości funkcji kosztu MSE od rzędu N dla wszystkich testowanych modeli



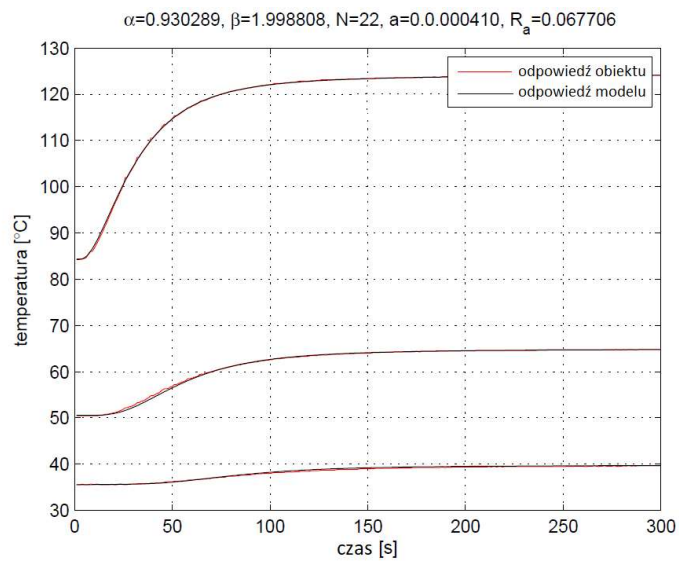
Rys. 3.10. Odpowiedź skokowa obiektu oraz modelu 1 z rzędem całkowitym ($\alpha = 1$, $\beta = 2$) oraz optymalnym rzędem $N = 10$, $a = 0.000806$, $R_a = 0.034214$



Rys. 3.11. Odpowiedź skokowa obiektu oraz modelu 2 z rzędem niecałkowitym ($\alpha = 0.931555$, $\beta = 2$) oraz optymalnym rzędem $N = 22$, $a = 0.000419$, $R_a = 0.066532$



Rys. 3.12. Odpowiedź skokowa obiektu oraz modelu 3 z rzędem niecałkowitym ($\alpha = 1$, $\beta = 2.089926$) oraz optymalnym rzędem $N = 14$, $a = 0.000646$, $R_a = 0.034243$



Rys. 3.13. Odpowiedź skokowa obiektu oraz modelu 4 z rzędem niecałkowitym ($\alpha = 0.930289, \beta = 1.998808$) oraz optymalnym rzędem $N = 22$, $a = 0.000410, R_a = 0.0677066$

Do estymacji parametrów wykorzystano znaną funkcję kosztu MSE. Oszacowano również optymalny rząd wszystkich proponowanych modeli w przestrzeni stanów.

Wnioski z porównania badanych ciągłych modeli w przestrzeni stanu eksperymentalnego obiektu można sformułować następująco:

- W artykule przedstawiono problem identyfikacji parametrów dla niecałkowitego rzędu modeli w przestrzeni stanów procesu wymiany ciepła.
- Porównano 4 różne modele - pierwszy klasyczny - rzędu całkowitego $\alpha = 1, \beta = 2$ oraz 3 modele niecałkowitego rzędu $\alpha = 1, \beta \in \mathbb{R}$ i $\alpha \in \mathbb{R}, \beta = 2$ i $\alpha \in \mathbb{R}, \beta \in \mathbb{R}$.
- Wszystkie proponowane modele niecałkowitego rzędu są dokładniejsze w sensie funkcji kosztu MSE niż znany model rzędu całkowitego.
- Najlepszym modelem pod względem minimalizacji funkcji kosztu MSE jest model o niecałkowitym rzędzie ($\alpha, \beta \in \mathbb{R}$). Ten model jest najdokładniejszy, ale wolniej zbieżny niż pozostałe modele.
- Najszybszą zbieżność osiąga model rzędu całkowitego ($\alpha = 1, \beta = 2$), ale jest on najmniej dokładny.
- We wszystkich testowanych modelach dokładność modelu zależy od rzędu aproksymacji N - im wyższy rząd N , tym lepsza dokładność.
- Dla każdego z modeli można wyznaczyć optymalny rząd aproksymacji N :
 - $\alpha = 1, \beta = 2 - N = 10$
 - $\alpha = 1, \beta \in \mathbb{R} - N = 22$
 - $\alpha \in \mathbb{R}, \beta = 2 - N = 14$
 - $\alpha \in \mathbb{R}, \beta \in \mathbb{R} - N = 22$
- W dalszych badaniach będzie testowana stabilność, sterowalność oraz obserwowalność tych modeli.

3.2. Dyskretny model niecałkowitego rzędu w przestrzeni stanu

Rozważymy teraz model dyskretny naszego obiektu cieplnego niecałkowitego rzędu w przestrzeni stanu.

Przy użyciu definicji różnicy wstecznej ułamkowego rzędu (1.53) dyskretne równanie stanu ułamkowego rzędu jest określone w następujący sposób (zobacz [77]):

$$\begin{cases} (\Delta_L^\alpha x)(t+h) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \quad (3.35)$$

gdzie $x(t) \in \mathbb{R}^N$ jest wektorem stanu, $u(t) \in \mathbb{R}^R$ jest sterowaniem, $y(t) \in \mathbb{R}^M$ jest wyjściem. A , B i C są odpowiednio macierzami stanu, sterowania i wyjścia. Jeśli krótko oznaczymy k -ty moment czasowy $h * k$ przez k , to następnie równanie (3.35) zmienia się na:

$$\begin{cases} (\Delta_L^\alpha x)(k+1) = A^+x(k) + B^+u(k) \\ y(k) = C^+x(k). \end{cases} \quad (3.36)$$

gdzie:

$$A^+ = h^\alpha A. \quad (3.37)$$

$$B^+ = h^\alpha B. \quad (3.38)$$

$$C^+ = C. \quad (3.39)$$

Skończona wersja rozwiązania równania stanu (3.36) przy użyciu dykretnego operatora GL zdefiniowanego wzorem (1.8) przybiera postać:

$$x(k+1) = (h^\alpha A - c_1 I)x(0) + \sum_{l=1}^L c_{l+1} x(k-l) + h^\alpha Bu(k) \quad (3.40)$$

gdzie:

$$G^+ = A^+ + \alpha I. \quad (3.41)$$

$$A_l^+ = d_l I_{N \times N}. \quad (3.42)$$

W tym miejscu wprowadzimy pojęcie praktycznej stabilności. Zostało ono zaprezentowane przez Prof. T. Kaczorka w [78], rozważano też w [79] oraz [80]. Wiąże on stabilność dyskretnego układu niecałkowitego rzędu opisanego równaniem stanu (3.36) z asymptotyczną stabilnością jego aproksymowanego rozwiązania podanego równaniem (3.40). Prezentowane poniżej warunki stabilności zostały opublikowane w [29].

Definicja 21. (*Stabilność praktyczna*)

System rzędu ułamkowego opisany przez (3.36) jest praktycznie stabilny, jeśli jego skończone wymiarowe rozwiązanie (3.40) jest asymptotycznie stabilne.

Jeśli dodatkowo założymy, że rozważany system dyskretny FO (3.36) jest dodatni, to można zastosować proste warunki praktycznej stabilności. Warunki te podane są w [79], [80]. W tym artykule zostaną zastosowane (Twierdzenia 3 i 5 w [79]):

Propozycja 2. (*Konieczny i wystarczający warunek praktycznej stabilności układu dodatniego (3.36) dla stałej długości pamięci L*)

Dodatni układ niecałkowitego rzędu (3.36) o rzędzie $0 < \alpha < 1$ jest praktycznie stabilny wtedy i tylko wtedy, gdy standardowy układ dodatni:

$$x(k+1) = (G^+ + \sum_{l=2}^L A_l)x(k). \quad (3.43)$$

jest asymptotycznie stabilny.

Propozycja 3. (*Konieczny i wystarczający warunek praktycznej stabilności układu dodatniego (3.36) niezależnie od długości pamięci*)

Dodatni układ niecałkowitego rzędu (3.36) z rzędem $0 < \alpha < 1$ jest praktycznie stabilny dla każdej długości pamięci L wtedy i tylko wtedy, gdy standardowy układ dodatni:

$$x(k+1) = (A^+ + I)x(k). \quad (3.44)$$

jest asymptotycznie stabilny.

Oba powyższe twierdzenia zostaną zastosowane do oszacowania stabilności dyskretnego modelu przewodnictwa cieplnego rozpatrywanego w tej rozprawie.

Model dyskretny wywodzi się bezpośrednio z modelu ciągłego (3.20) po zastosowaniu dyskretnej wersji definicji GL (1.8). Jego rozwiązanie (3.40) ma postać:

$$\begin{cases} Q^+(k+1) = G^+Q^+(k) - \sum_{l=1}^L A_l^+Q^+(k-l) + B^+u(k) \\ y^+(k) = C^+Q^+(k) \end{cases} \quad (3.45)$$

W (3.45) A_l^+ jest wyrażone przez (3.42), G^+ , B^+ i C^+ mają następującą postać:

$$\begin{cases} G^+ = \text{diag}\{\lambda_{\beta_1}^+, \lambda_{\beta_2}^+ \dots \lambda_{\beta_N}^+\}. \\ B^+ = h^\alpha B. \\ C^+ = C. \end{cases} \quad (3.46)$$

gdzie:

$$\lambda_{\beta_n}^+ = \alpha - h^\alpha (a\pi^\beta n^\beta + R_a). \quad (3.47)$$

W pracy [20] wykazano, że widmo układu ciągłego w czasie można rozłożyć na pojedyncze, oddzielne wartości własne (analogicznie jak w przypadku rzędu całkowitego). Ta własność jest również odwzorowywana w systemie czasu dyskretnego. W szczególności rozwiązanie (3.45) można również rozłożyć na oddzielne „podrozwiązania” związane z pojedynczymi wartościami własnymi (3.47). Pozwala to na podanie fundamentalnych wyników przedstawionych poniżej.

Prezentację głównych wyników rozpoczniemy od dekompozycji modelu dyskretnego (3.45). Wektor stanu $Q^+(k)$ można zapisać jako:

$$Q^+(k) = \begin{bmatrix} q_1^+(k) \\ \dots \\ q_N^+(k) \end{bmatrix}. \quad (3.48)$$

Macierze G^+ i A_l^+ opisujące rozwiązanie układu dyskretnego (3.45) są macierzami diagonalnymi. W konsekwencji rozwiązanie (3.45) można rozłożyć na N niezależnych modów, powiązanych z n -tą zmienną stanu $q_n^+(k)$ i opisanych przez n wartości własnych. Analizę stabilności można przeprowadzić za pomocą rozwiązania swobodnego. Dla stałej długości pamięci L przyjmuje postać:

$$q_n^+(k+1) = \lambda_{\beta_n}^+ q_n^+(k) - \sum_{l=2}^L d_l q_n^+(k-l), \quad n = 1, \dots, N. \quad (3.49)$$

Wielomian charakterystyczny związany z rozwiązaniem $q_n^+(k)$ ma postać:

$$w_n(z^{-1}) = 1 - \lambda_{\beta_n}^+ z^{-1} + \sum_{l=2}^L d_l z^{-l}. \quad (3.50)$$

Dla każdej długości pamięci rozwiązanie przyjmuje postać:

$$q_n^+(k+1) = \lambda_{\beta_n}^+ q_n^+(k) - \sum_{l=2}^{\infty} d_l q_n^+(k-l), \quad n = 1, \dots, N. \quad (3.51)$$

i analogicznie wielomian charakterystyczny ma postać:

$$w_n(z^{-1}) = 1 - \lambda_{\beta_n}^+ z^{-1} + \sum_{l=2}^{\infty} d_l z^{-l}. \quad (3.52)$$

gdzie $\lambda_{\beta_n}^+$ jest wyrażone przez (3.47), d_l przez (1.51) lub przez (1.52).

Zauważmy, że praktyczna stabilność lub niestabilność całego rozważanego systemu jest określona przez asymptotyczną stabilność lub niestabilność jego oddzielnych modów (3.49) lub (3.51). Można to wyrazić następującymi uwagami:

Uwaga 1. (Praktyczna stabilność dyskretnego, zdekomponowanego systemu niecałkowitego rzędu)

1. Dyskretny system niecałkowitego rzędu (3.45) będzie praktycznie stabilny dla stałej długości pamięci L wtedy i tylko wtedy, gdy każdy mod jego rozwiązania (3.49) jest asymptotycznie stabilny.
2. Dyskretny system niecałkowitego rzędu (3.35) będzie praktycznie stabilny dla każdej długości pamięci L wtedy i tylko wtedy, gdy każdy mod jego rozwiązania (3.51) jest asymptotycznie stabilny.

Uwaga 2. (Niestabilność dyskretnego, zdekomponowanego systemu niecałkowitego rzędu)

1. Dyskretny system niecałkowitego rzędu (3.45) będzie niestabilny dla stałej długości pamięci L wtedy i tylko wtedy, gdy istnieje co najmniej jeden niestabilny mod jego rozwiązania (3.49).
2. Dyskretny system niecałkowitego rzędu (3.35) będzie praktycznie niestabilny dla każdej długości pamięci L wtedy i tylko wtedy, gdy istnieje co najmniej jeden niestabilny mod jego rozwiązania (3.51).

Praktyczną stabilność systemu dyskretnego, z którym mamy do czynienia, można bezpośrednio przetestować za pomocą obu powyższych uwag. Wymaga to od nas przetestowania lokalizacji pierwiastków każdego charakterystycznego wielomianu (3.50) dla $n = 1, \dots, N$. Stopień każdego wielomianu jest równy $L + 1$. Sprawia to, że korzystamy wyłącznie z metod numerycznych i można to zrobić z pomocą Matlab. Przedstawiony zostanie również przykład takiego testu stabilności.

Jednak dekompozycja równania stanu i możliwość testowania stabilności poprzez badanie rozproszonych systemów skalarnych N pozwalają na sformułowanie analitycznych warunków stabilności. W tym celu zostaną użyte Twierdzenia 2 i 3. Zanim zaczniemy, należy krótko omówić pozytywność rozważanego systemu.

Na początku należy zauważyć, że analiza stabilności wymaga od nas rozważenia tylko zachowania stanu $Q^+(k)$, wejścia i wyjścia systemu, opisane operatorami B i C nie wymagają analizy. Od razu można zauważyć, że operatorem stanu dla układu ciągłego w czasie (3.6), (3.7) jest macierz Metzlera (definicja macierzy Metzlera podana jest np. w [79]). Oznacza to, że ciągły w czasie stan naszego układu jest dodatni i asymptotycznie stabilny.

Twierdzenia 2 i 3 są formułowane przy użyciu systemów standardowych. Można je łatwo skonstruować dla każdego modu naszego zdekomponowanego systemu z osobna. n -ty zdekomponowany, standardowy system ma postać dla stałej długości pamięci L :

$$q_n^+(k+1) = \left(\lambda_{\beta_n}^+ + \sum_{l=2}^L d_l \right) q_n^+(k), \quad n = 1, \dots, N. \quad (3.53)$$

System (3.53) jest systemem skalarnym pierwszego rzędu z jedną wartością własną $\lambda_{sL_n}^+$:

$$\lambda_{sL_n}^+ = \lambda_{\beta_n}^+ + \sum_{l=2}^L d_l. \quad (3.54)$$

i dla każdej długości pamięci:

$$q_n^+(k+1) = (h^\alpha \lambda_{\beta_n} + 1) q_n^+(k), \quad n = 1, \dots, N. \quad (3.55)$$

Wartość własna układu (3.55) jest równa:

$$\lambda_{s_n}^+ = h^\alpha \lambda_{\beta_n} + 1. \quad (3.56)$$

n -ty mod zdekomponowanego systemu (3.53) lub (3.55) będzie asymptotycznie stabilny wtedy i tylko wtedy, gdy: $|\lambda_{sL_n}^+| < 1$ lub $|\lambda_{s_n}^+| < 1$. Cały system będzie stabilny wtedy i tylko wtedy, gdy wszystkie jego mody są asymptotycznie stabilne. Pozwala to na sformułowanie warunków analitycznych wiążących praktyczną stabilność z rozmiarem modelu N , czasem próbkowania h i długością pamięci L . Warunki te są formułowane następująco:

Propozycja 4. (Maksymalny rozmiar modelu N zapewniający praktyczną stabilność modelu dyskretnego dla stałej długości pamięci L)

1. rozważamy dyskretny model procesu wymiany ciepła opisany przez (3.45),
2. rozwiązanie modu n zdekomponowanego systemu wyraża się wzorem (3.49),
3. standardowy system niecałkowitego rzędu powiązany z n -tym modem ma postać (3.53).

Wielkość N skończenie wymiarowej aproksymacji zapewniającej praktyczną stabilność modelu dyskretnego (3.45) spełnia następującą nierówność:

$$N \leq \text{Int} \left(\left(\left(\frac{1 + \alpha - h^\alpha R_a + \sum_{l=2}^{L+1} d_l}{h^\alpha a \pi^\beta} \right)^{\frac{1}{\beta}} \right) \right). \quad (3.57)$$

Propozycja 5. (Maksymalny rozmiar modelu N zapewniający praktyczną stabilność modelu dyskretnego dla każdej długości pamięci)

Założenia:

1. rozważamy dyskretny model procesu wymiany ciepła opisany przez (3.45),
2. rozwiązanie modu n zdekomponowanego systemu jest wyrażone przez (3.51),
3. standardowy system niecałkowitego rzędu powiązany z n -tym modem ma postać (3.55).

Maksymalny wymiar N skończenie wymiarowej aproksymacji zapewniającej praktyczną stabilność modelu dyskretnego (3.45) spełnia następującą nierówność:

$$N \leq \text{Int} \left(\left(\frac{2 - h^\alpha R_a}{h^\alpha a \pi^\beta} \right)^{\frac{1}{\beta}} \right). \quad (3.58)$$

Innym ważnym problemem podczas stosowania i implementacji rozważanego dyskretnego modelu niecałkowitego rzędu procesu wymiany ciepła jest oszacowanie wartości czasu próbkowania h niezbędnego do zachowania praktycznej stabilności modelu dla ustalonego, skończonego rzędu N . Zwiększenie N poprawia dokładność modelu. Może to zachęcić do używania wysokiego rzędu N niezależnie od innych ograniczeń. Wtedy rozwiązaniem może być krótszy czas próbkowania h . Odpowiednie warunki można również sformułować za pomocą odpowiednich oszacowań:

Propozycja 6. (Maksymalny okres próbkowania h zapewniający praktyczną stabilność modelu dyskretnego dla stałej długości pamięci L)

Założenia:

1. rozważamy dyskretny model procesu wymiany ciepła opisany przez (3.45),
2. rozwiązanie modu n zdekomponowanego systemu jest wyrażone przez (3.51),
3. standardowy system niecałkowitego rzędu powiązany z n -tym modem ma postać (3.55).

Maksymalny rozmiar h okresu próbkowania zapewniający praktyczną stabilność modelu dyskretnego (3.45) spełnia następującą nierówność:

$$h < \left(\frac{1 + \alpha + \sum_{l=2}^L d_l}{a \pi^\beta n^\beta + R_a} \right)^{\frac{1}{\alpha}}. \quad (3.59)$$

Propozycja 7. (Maksymalny okres próbkowania h zapewniający praktyczną stabilność modelu dyskretnego dla każdej długości pamięci L)

Założenia:

1. rozważamy dyskretny model procesu wymiany ciepła opisany przez (3.45),
2. rozwiązanie modu n zdekomponowanego systemu jest wyrażone przez (3.51),
3. standardowy system niecałkowitego rzędu powiązany z n -tym modem ma postać (3.55).

Maksymalny rozmiar h okresu próbkowania zapewniający praktyczną stabilność modelu dyskretnego (3.45) spełnia następującą nierówność:

$$h < \left(\frac{2}{a\pi^\beta n^\beta + R_a} \right)^{\frac{1}{\alpha}}. \quad (3.60)$$

Analogicznie jak powyżej warunek (3.60) jest przypadkiem granicznym warunku (3.59).

Rozważmy przykład obiektu cieplnego z rozdziału 2. Liczbowe wartości parametrów obiektu obliczono numerycznie (patrz [20]) z wykorzystaniem funkcji kosztu MSE. Są one podane w tabeli 3.6.

Tabela 3.6. Parametry obiektu cieplnego

Parametr	α	β	a	R_a
wartość	0.9430	1.9847	0.0005	0.0591

Dla układu z parametrami podanymi w tabeli 3.6 maksymalne wymiary modelu N dla różnych h oszacowane względem warunków (3.57) - (3.58) podano w tabeli 3.7. Maksymalne wartości okresu próbkowania h dla stałych wartości wymiaru modelu N podano w tabeli 3.7.

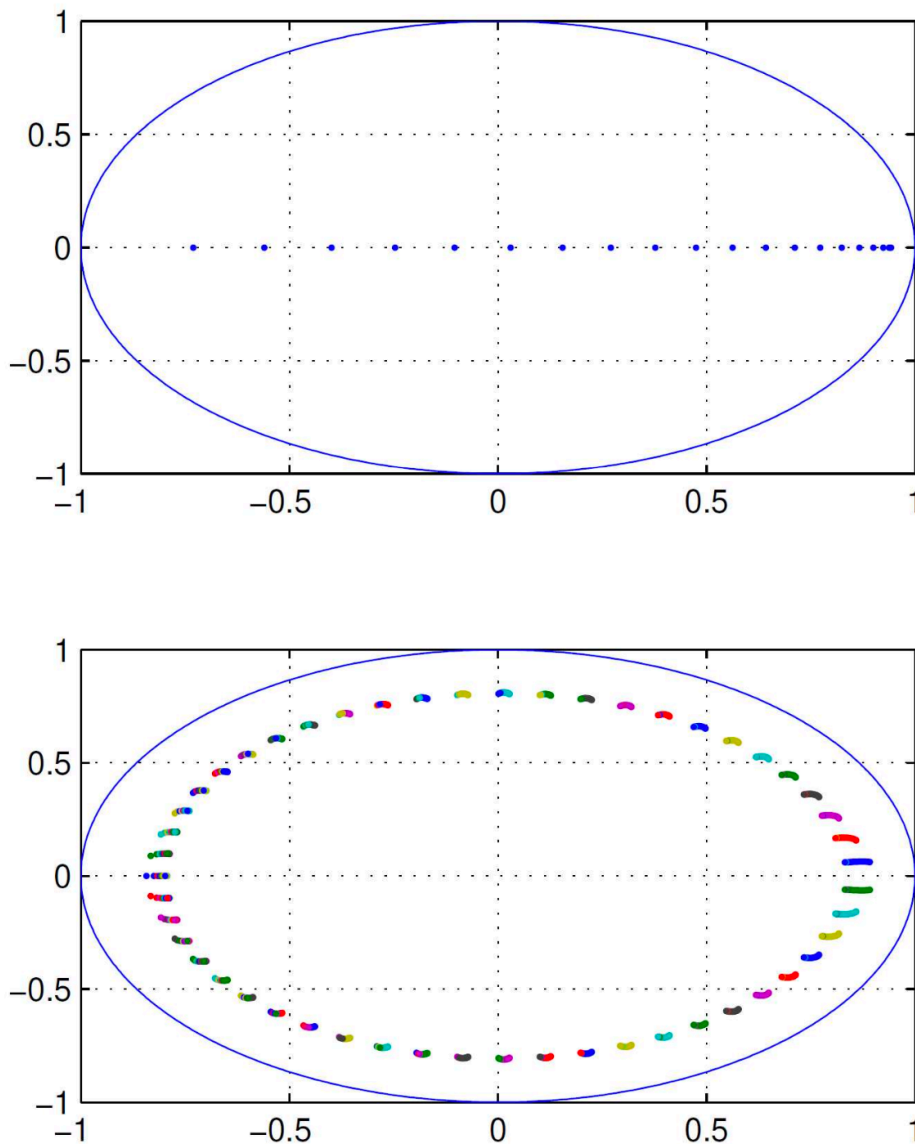
Tabela 3.7. Maksymalny wymiar modelu N dla różnych okresów próbkowania h

Okres próbkowania h [s]	0.5	1	2	5
N_{max} dla $L=50$	29	20	15	9
N_{max} dla każdego L	29	20	15	9

Aby zweryfikować powyższe wyniki na rysunkach 3.14 i 3.15 pokazano przykładowe widma układów standardowych, obliczone jako zbiory wartości własnych (3.54). Aby zweryfikować wyniki widma całego dyskretnego systemu obliczono pierwiastki wielomianów charakterystycznych (3.50).

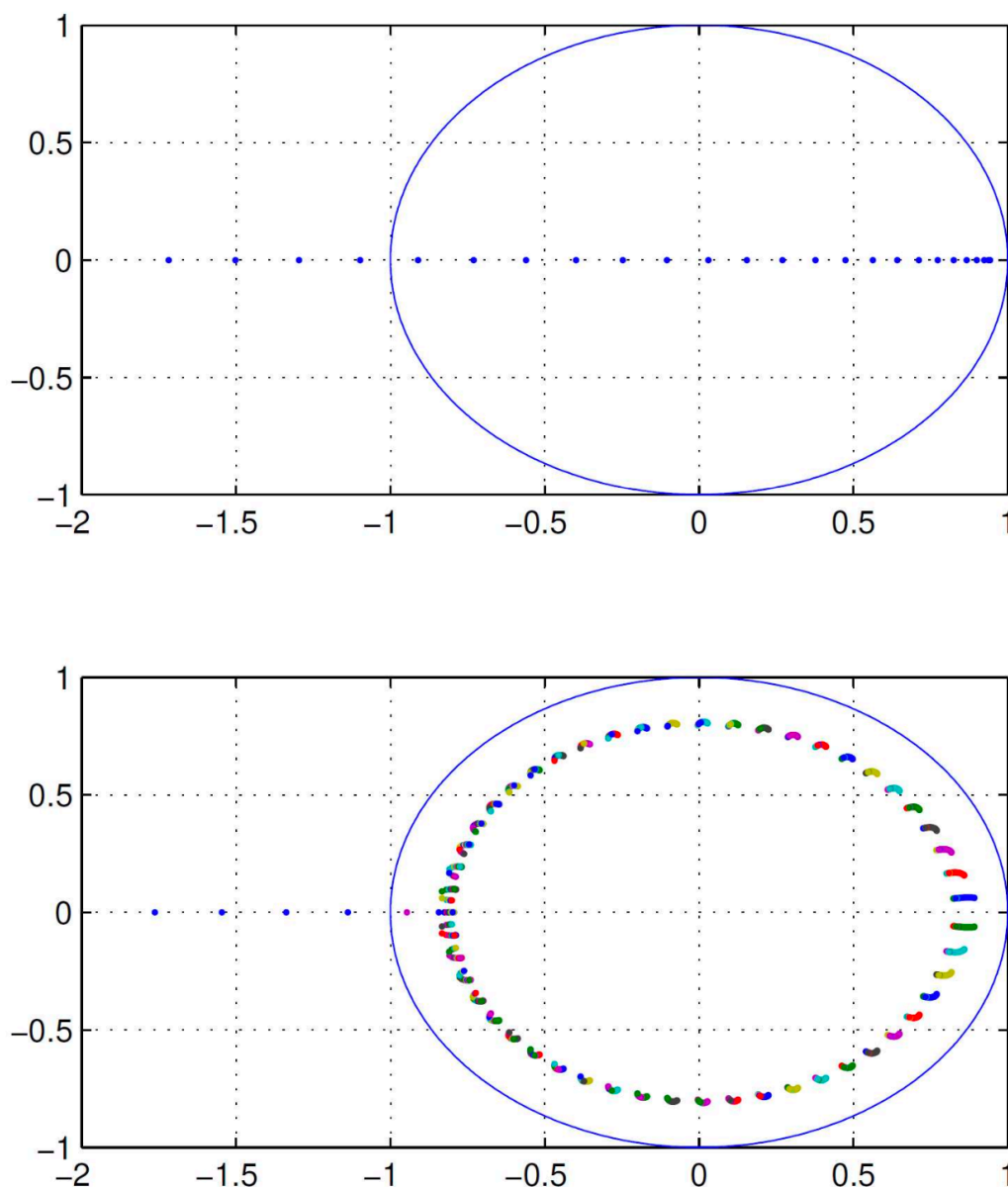
Tabela 3.8. Maksymalny wymiar okresu próbkowania h dla ustalonego wymiaru modelu N

Wymiar modelu N	10	20	30	40
h_{max} [s] dla $L=50$	3.7971	1.0471	0.4764	0.2708
h_{max} [s] dla każdego L	3.7934	1.0461	0.4760	0.2706



Rys. 3.14. Widma układów stabilnych: $h=1$ s, $N=20$.

Oba diagramy 3.14 i 3.15 potwierdzają poprawność zaproponowanych warunków stabilności.



Rys. 3.15. Widma układów niestabilnych: $h=1$ s, $N=25$.

Rozważmy jeszcze możliwość zastosowania skończonego wymiarowego dyskretnego modelu dla potrzeb cyfrowej implementacji w sterowniku PLC.

Równania (3.36) opisują nieskończone wymiarowy model, wynikający bezpośrednio z równania (3.20). Jego skończone wymiarowa aproksymacja wymiaru N^+ jest opisana analogicznie, ale operatory (3.46) są interpretowane jako macierze. Rozwiązanie dyskretnego równania stanu można obliczyć bezpośrednio za pomocą wzoru podanego przez równanie (3.45).

Cyfrowa implementacja modelu (3.45) wymaga od nas ustawienia długości pamięci L , tak aby zapewniona była żądana dokładność. Jego sensowna ocena jest kluczowym problemem podczas cyfrowej implementacji modeli niecałkowitego rzędu. Zazwyczaj ten parametr powoduje problemy podczas implementacji, ponieważ musi być stosunkowo duży. Niestety rząd rozpatrywanego modelu jest również wysoki ($N = 22$ dla przypadku ciągłego). Oznacza to, że oszacowanie rozsądnych rzędów N^+ oraz L przed cyfrową implementacją jest ważnym zagadnieniem. Można go rozwiązać numerycznie za pomocą podejścia przedstawionego na przykład w artykule [20].

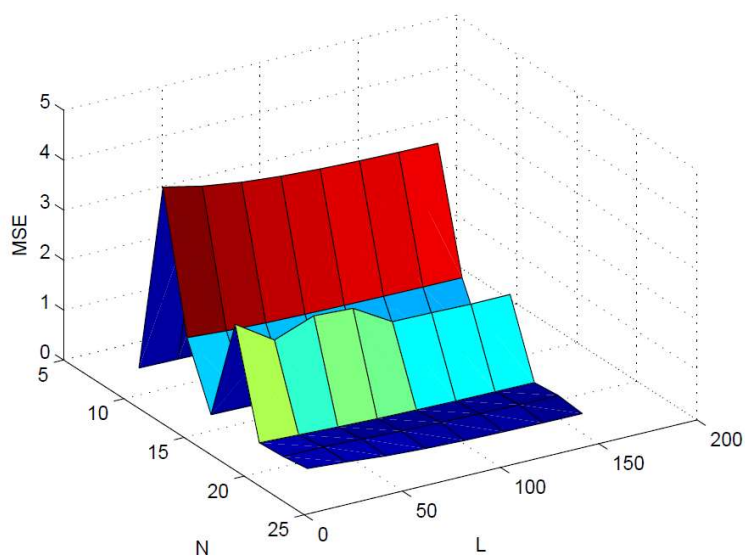
Eksperymenty zostały wykonane przy użyciu systemu przedstawionego w rozdziale 2. Dyskretny model (3.45) został zbadany przy użyciu funkcji kosztu MSE (3.34).

Celem badań było oszacowanie wartości wymiaru skończonego modelu N i długości pamięci L , zapewniających akceptowalną dokładność prezentowanego, dyskretnego modelu niecałkowitego rzędu. Aby to zrobić, obliczono wartości funkcji kosztu (3.34) dla różnych wartości N i L , a następnie wykreślono wykres 3D - rysunek 3.16.

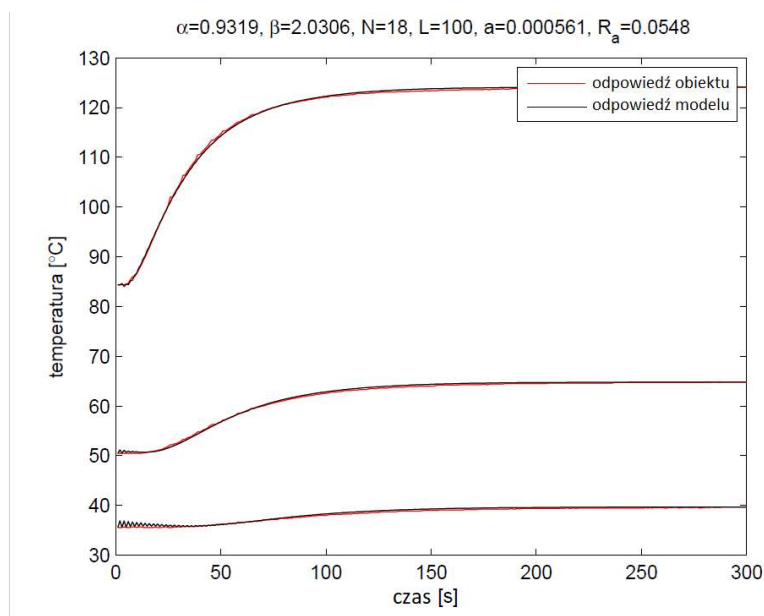
Wyniki przedstawiono w tabeli 3.9 oraz na rysunku 3.17.

Tabela 3.9. Funkcja kosztu MSE (3.34) dla różnych rzędów N i wszystkich testowanych modeli

L	20	40	60	80	100	120	140	160
8	0.1134	0.1027	0.0901	0.0794	0.0714	0.0655	0.0612	0.0582
10	4.0430	3.8739	3.7606	3.6906	3.6906	3.6193	3.6011	3.5890
12	1.3441	1.2896	1.2591	1.2401	1.2282	1.2207	1.2160	1.2130
14	0.1137	0.1026	0.0868	0.0733	0.0632	0.0560	0.0508	0.0472
16	2.2272	1.7243	2.0186	1.9760	1.5203	1.5110	1.4624	1.4975
18	0.1680	0.1365	0.1048	0.0823	0.0669	0.0565	0.0493	0.0444
20	0.2011	0.2059	0.1716	0.1446	0.1252	0.1115	0.1017	0.0948
22	0.2711	0.2157	0.1443	0.0994	0.0742	0.0582	0.0484	0.0426



Rys. 3.16. Zależność funkcji kosztu MSE (3.34) dla różnych rzędów N i wszystkich testowanych modeli



Rys. 3.17. Odpowiedź skokowa obiektu oraz modelu z niecałkowitym rzędem ($\alpha = 0.9319$, $\beta = 2.0306$) oraz optymalnego rzędu $N = 18$, $L = 100$, $a = 0.000561$, $R_a = 0.0548$

Wyniki przedstawione w tym rozdziale można zastosować podczas cyfrowej implementacji omawianego dyskretnego modelu rzędu niecałkowitego w sterowniku PLC. Taki model zawsze jest kompromisem pomiędzy dokładnością a możliwościami platformy cyfrowej. Podejście zaproponowane w tym rozdziale można również uogólnić na inne systemy niecałkowitego rzędu opisane równaniem stanu z diagonalną macierzą stanu. Taki model można uzyskać poprzez transformację do postaci kanonicznej Jordana.

3.3. Ciągły model w postaci transmitancji hybrydowej niecałkowitego rzędu

Wyniki zawarte w [69] pokazują, że dobry w sensie dokładności model niskiego rzędu można budować jako połączenie funkcji całkowego oraz ułamkowego rzędu. Do modelowania obiektów o wysokim rzędzie proponuje się następujące modele (patrz [19]):

$$G_1(s) = \frac{1}{(T_{\alpha_1} s^{\alpha_1} + 1)(T_{n_1} s + 1)^{n_1}} \quad (3.61)$$

gdzie: T_{α_1} oraz T_{n_1} oznaczają stałe czasowe ułamkowego oraz całkowego rzędu, $0 < \alpha_1 < 1$ oraz $n_1 \in \mathbb{Z}$ oznaczają kolejno ułamkowy i całkowity rząd transmitancji. Powyższy model został przedstawiony w pracy [69]. Element ułamkowego rzędu s^{α_1} w transmitancji (3.61) można przybliżać za pomocą aproksymacji ORA, wyrażonej przez (1.35) - (1.40).

Alternatywny model proponowany jest następująco:

$$G_2(s) = \frac{1}{(T_{\alpha_2} s + 1)^{\alpha_2} (T_{n_2} s + 1)^{n_2}} \quad (3.62)$$

gdzie: T_{α_2} oraz T_{n_2} oznaczają również stałe czasowe rzędu ułamkowego oraz całkowego, $0 < \alpha_2 < 1$ oraz n_2 oznaczają kolejno rząd ułamkowy i całkowity transmitancji. Elementarny element inercyjny rzędu ułamkowego $\frac{1}{(T_{\alpha_2} s + 1)^{\alpha_2}}$ we wzorze (3.62) jest przybliżany za pomocą aproksymacji Charefa (1.43)-(1.49).

Odpowiedź skokowa każdej powyższej aproksymacji jest opisana poniżej:

$$y_{1,2}(t) = \mathcal{L}^{-1}\left\{\frac{1}{s}G_{1,2}(s)\right\} \quad (3.63)$$

We wzorze (3.63) $G_{1,2}(s)$ jest opisane przez (3.61) lub (3.62).

Dokładność powyższych proponowanych modeli można oszacować za pomocą następującej funkcji kosztu:

$$I_m = \int_0^{\infty} (y(t) - y_m(t)) dt \quad (3.64)$$

We wzorze (3.64) $y(t)$ oznacza odpowiedź skokową rzeczywistego obiektu, a $y_m(t)$ oznacza odpowiedź skokową odpowiedniego rozpatrywanego modelu.

Ważnym problemem podczas korzystania z obu proponowanych modeli może być skrócenie się aproksymowanych zer, wyrażonych przez (1.36) lub (1.47) przez bieguny części całkowitej rzędu równego $\frac{1}{T_{n_1}}$ lub $\frac{1}{T_{n_2}}$ odpowiednio. To znacznie zmniejsza dokładność modelu i należy tego unikać. Analizę podobnego problemu przeprowadzono w pracy [81].

Następnie zostanie pokazane użycie obu proponowanych modeli hybrydowych (3.61) i (3.62) do modelowania rozpatrywanego obiektu cieplnego.

Parametry tych modeli minimalizujące funkcję kosztu (3.64) zostały obliczone przy użyciu pakietu MATLAB, funkcją *fminsearch*. Otrzymane parametry podane są w tabelach - dla modelu (3.61) w tabeli 3.10 oraz dla modelu (3.62) w tabeli 3.11.

Tabela 3.10. Parametry modelu (3.61)

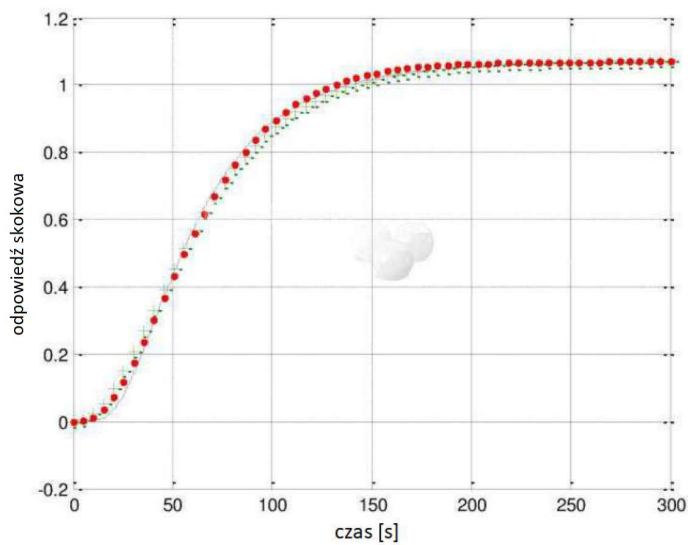
Parametr	n_1	α_1	T_{α_1}	T_{n_1}	I_1	ω_l	ω_h
Wartości	2	-0.5779	426.0690	21.7087	0.1306	0.1	10
Wartości	3	19.5602	96.959	22.1088	0.0560	0.1	10

Tabela 3.11. Parametry modelu (3.62)

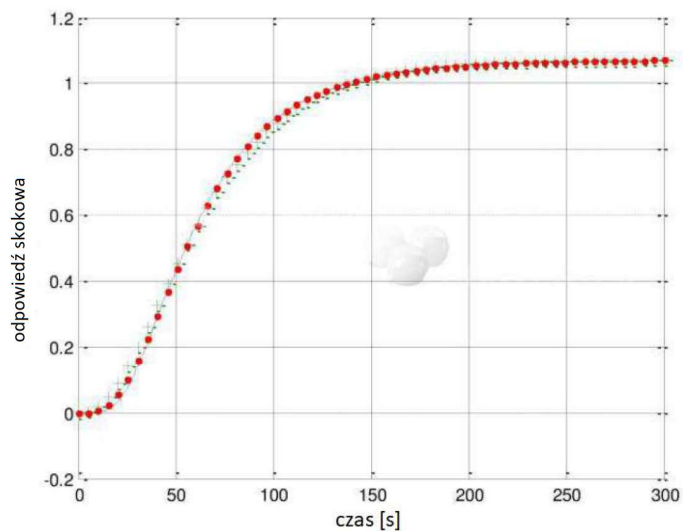
Parametry	n_2	α_2	T_{α_2}	T_{n_2}	I_2	Δ
Wartości	2	0.6628	32.3744	24.6591	0.0995	0.5
Wartości	3	0.4773	55.6602	14.6059	0.0091	0.5

Odpowiedź skokowa modelu (3.61) pokazana jest na rysunku 3.18, odpowiedź skokowa modelu (3.62) jest pokazana na rysunku 3.19.

Porównanie wszystkich omawianych modeli transmitancji całkowitych oraz ułamkowych przy pomocy funkcji kosztu (3.64) oraz podsumowanie ich rzędów aproksymacji jest zebrane w tabeli 3.12.



Rys. 3.18. Odpowiedź skokowa: obiektu (linia ciągła), modelu (3.61) z $n_1 = 2$ (oznaczone jako "+") oraz modelu (3.61) z $n_1 = 3$ (oznaczone ".".)



Rys. 3.19. Odpowiedź skokowa: obiektu (linia ciągła), modelu (3.62) z $n_2 = 2$ (oznaczone jako "+") oraz modelu (3.62) z $n_2 = 3$ (oznaczone ".".)

Tabela 3.12. Porównanie modeli całkowitego oraz ułamkowego rzędu

Parametr	funkcja kosztu (3.64)	rzęd
model w przestrzeni stanu (2.2)-(2.12)	0.0156	25
transmitancja Kupfmüllera (2.13)	0.0039	12
transmitancja Strejca (2.15)	0.1514	3
Proponowany model ułamkowego rzędu (3.61)	0.0560	8
Proponowany model ułamkowego rzędu (3.62)	0.0091	8

Z tabeli 3.10 możemy wywnioskować, że rząd całkowity $n_1 = 3$ dla modelu (3.61) zapewnia lepszą dokładność modelu w sensie funkcji kosztu (3.64) niż model z $n_1 = 2$.

Wniosek z tabeli 3.10 jest również analogiczny - wyższy rząd całkowity $n_2 = 3$ dla modelu (3.62) daje dokładniejszy model w sensie funkcji kosztu (3.64) niż model z rzędem $n_2 = 2$.

Model inercjalny (3.62) jest nieco dokładniejszy niż model (3.61) zawierający elementarny element różniczkowo-inercyjny.

Istotnym problemem podczas użytkowania obu proponowanych modeli hybrydowych jest problem skracania się biegunów aproksymacji z zerami części rzędu całkowitego. Może się to pojawić, gdy biegun użytego przybliżenia jest równy biegunowi części rzędu całkowitego. Zjawisko to powoduje utratę dokładności modelu i należy go unikać (patrz [81]).

Wyniki symulacji pokazują, że dokładność w sensie funkcji kosztu (3.64) obu proponowanych modeli hybrydowych ułamkowego rzędu jest dobra i zbliżona do modeli "klasycznych": równania stanu całkowitego rzędu oraz transmitancji drugiego rzędu z opóźnieniem. Jednocześnie - rząd obu proponowanych modeli hybrydowych jest znacznie mniejszy, niż rzędy modelu całkowitego w przestrzeni stanu oraz modeli transmitancji Kupfmüllera oraz Strejca.

3.4. Dyskretny model transmitancji niecałkowitego rzędu z opóźnieniem

Alternatywnym podejściem do modelowania rozważanego obiektu cieplnego, mniej złożonym od modeli niecałkowitego rzędu w przestrzeni stanu, jest zastosowanie modelu transmitancji z opóźnieniem. Transmitancja w przypadku całkowitego rzędu ma postać dobrze znanej transmitancji Kűpfmüllera, opisaną wzorem (2.14).

Modele transmitancji z opóźnieniem dla równania (2.14) można wyrazić w postaci dyskretniej, hybrydowej transmitancji $G^+(z^{-1})$ zawierającej całkowity oraz niecałkowity rząd. Całkowity rząd transmitancji reprezentowany jest jako opóźnienie, natomiast niecałkowity rząd modelowany jest przy użyciu aproksymacji CFE (1.56) jako inercja.

Proponowane modele możemy zapisać w postaci (wyniki opublikowane w [24]):

$$G_{FO1}(s) = \frac{ke^{-\tau_1 s}}{T_\alpha s^\alpha + 1} \quad (3.65)$$

$$G_{FO2}(s) = \frac{ke^{-\tau_2 s}}{(T_\alpha s^\alpha + 1)(T_\beta s^\beta + 1)} \quad (3.66)$$

Stosując aproksymację CFE (1.56) do transmitancji (3.65) oraz (3.66) oraz zakładając, że opóźnienia τ_1 i τ_2 są mnożnikami próby czasu h uzyskujemy:

$$G_{FO1}^+(z^{-1}) = \frac{z^{-N_1} Q_{\alpha M}(z^{-1})}{T_\alpha P_{\alpha M}(z^{-1}) + Q_{\alpha M}(z^{-1})} \quad (3.67)$$

$$G_{FO2}^+(z^{-1}) = \frac{z^{-N_2} Q_{\alpha M}(z^{-1}) Q_{\beta M}(z^{-1})}{(T_\alpha P_{\alpha M}(z^{-1}) + Q_{\alpha M}(z^{-1}))(T_\beta P_{\beta M}(z^{-1}) + Q_{\beta M}(z^{-1}))} \quad (3.68)$$

gdzie P oraz Q oznaczają licznik i mianownik aproksymacji CFE (1.56), M jest rzędem CFE oraz $\tau = N_{1,2}h$. Ważną zaletą stosowania CFE jest to, że wymaga ona użycia stosunkowo niskiego rzędu M (zwykle $M < 10$), aby uzyskać dobrą dokładność. Odpowiedzi skokowe obu modeli można obliczyć za pomocą funkcji Matlaba *step*.

Badania zostały wykonane przy użyciu systemu eksperymentalnego przedstawionego w rozdziale 2, dyskretny model czasu został zbadany przy użyciu funkcji kosztu, która w rozważanym aktualnie przypadku przyjmuje postać następującą ([75]):

$$MSE = \frac{1}{K_s} \sum_{k=1}^{K_s} (y(kh) - y^+(k))^2. \quad (3.69)$$

gdzie K_s jest liczbą wszystkich zebranych próbek ze wszystkich czujników, y jest eksperymentalną odpowiedzią zmierzoną w dyskretnych krokach czasowych kh , y^+ jest odpowiedzią czasową testowanego modelu, obliczoną wzdłuż tej samej siatki czasowej.

Parametry obu proponowanych modeli obliczane przy użyciu optymalizacji funkcji kosztu (3.69) są podane w tabelach 3.13 oraz 3.14.

Dodatkowo w tabelach 3.13 i 3.14 porównano wyniki z wynikami dla modeli rzędu całkowitego (2.14), uzyskanymi przy użyciu tej samej funkcji kosztu (3.69).

Dla wszystkich badań rząd aproksymacji był równy $M = 5$, czas próby był równy $h = 1s$. Porównanie odpowiedzi skokowych dla obiektu oraz dla każdego modelu niecałkowitego rzędu przedstawiono na rysunkach 3.20 oraz 3.21.

Tabela 3.13. Optymalne parametry i funkcja kosztów MSE (3.69) dla modelu (3.65) oraz dla modelu całkowitego rzędu (2.14)

Nr czujnika	$\alpha,$	$T_\alpha,$	$N_1,$	MSE	$T_1,$	$N_1,$	MSE
1	1.2183	38.0751	8	0.0944	29.0422	10	0.1133
2	1.5433	99.1564	16	0.0093	43.6374	23	0.0301
3	1.6967	195.5599	27	0.0033	82.7001	30	0.0182

Tabela 3.14. Optymalne parametry i funkcja kosztów MSE (3.69) dla modelu (3.66) oraz dla modelu całkowitego rzędu (2.14)

Nr czujnika	$\alpha,$	$\beta,$	T_α	T_β	$N_2,$	MSE	T_{21}	T_{22}	N_2	MSE
1	1.4704	0.7138	24.6555	43.8919	3	0.0434	28.3752	4.4113	6	0.0857
2	1.5703	0.8997	57.8378	48.3222	8	0.0043	40.1456	10.5969	15	0.0089
3	1.7666	1.1558	173.3362	64.1773	10	0.0021	60.6850	18.9781	26	0.0033

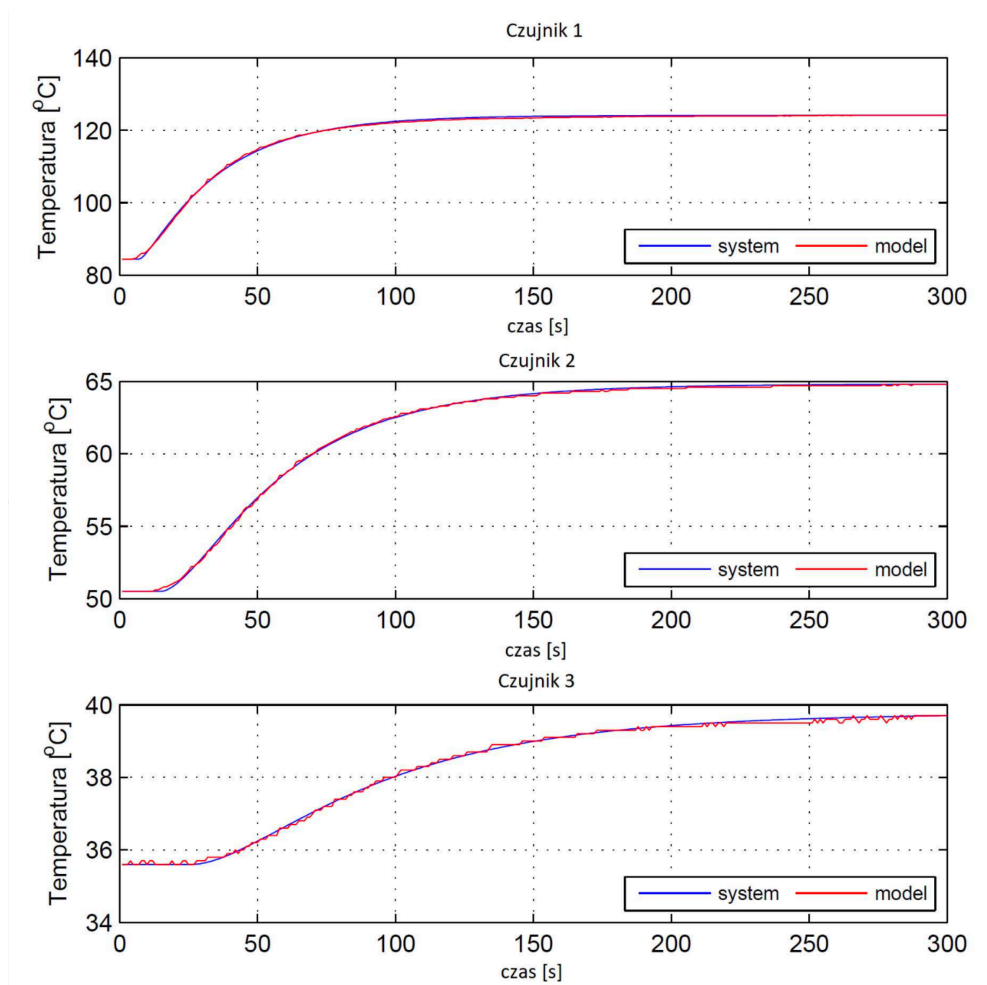
Z tabel 3.13 i 3.14 oraz rysunków 3.20 i 3.21 można wywnioskować, że zaproponowane dyskretne modele niecałkowitego rzędu są w stanie dokładniej opisać rozkład temperatury w rozważanym obiekcie doświadczalnym, niż znane modele transmitancji całkowitego rzędu z opóźnieniem.

Zaproponowane hybrydowe modele niecałkowitego rzędu z opóźnieniem są w stanie precyzyjnie opisać dynamikę rozważanego obiektu cieplnego (niższego rzędu i mniejszej wartości funkcji kosztu MSE niż odpowiadający im model rzędu całkowitego). Główną zaletą proponowanych modeli jest to, że dobrą dokładność modelu uzyskuje się dla jego stosunkowo niskiego rzędu - jest to suma rzędu przybliżonego CFE (równego 5) i rzędu całkowitej części opisującej opóźnienie jako wielokrotność czasu próbkowania.

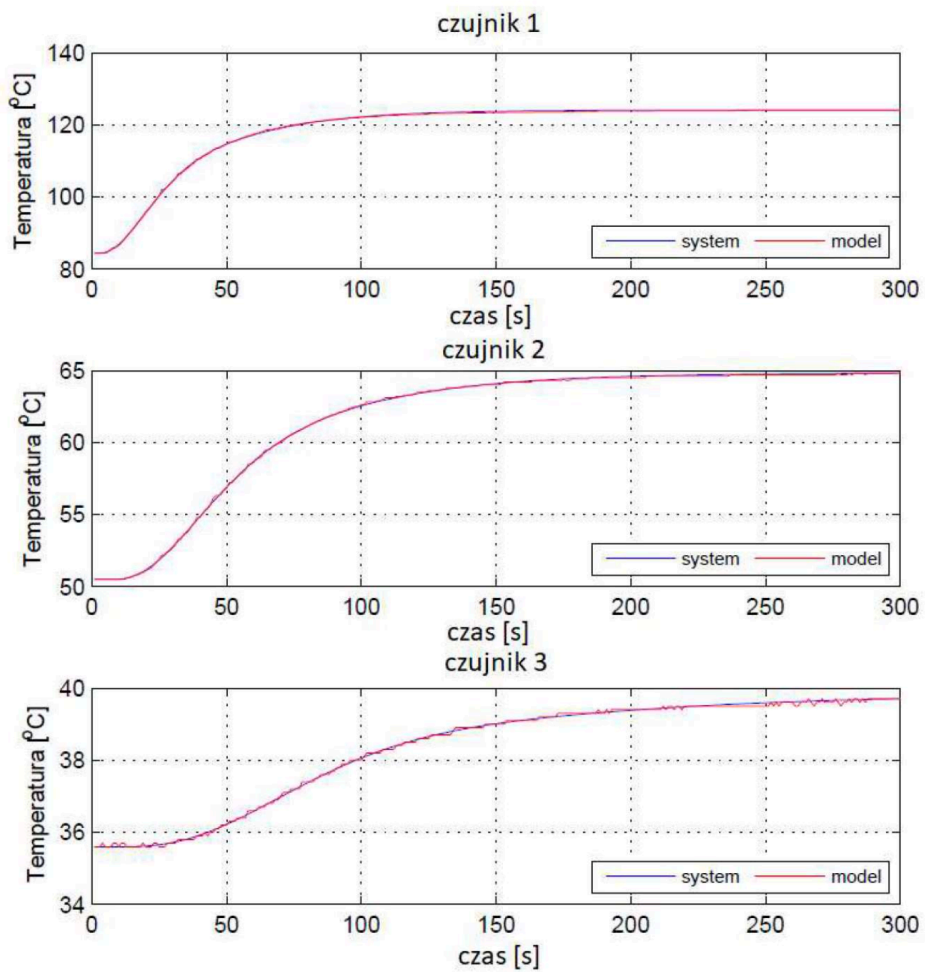
Kody źródłowe oprogramowania w środowisku Matlab, na podstawie których zostały wygenerowane powyższe wyniki, zostały umieszczone w Dodatku A.

Modele omówione w tym rozdziale zostały zaimplementowane w sterowniku PLC, a wyniki testów przedstawiono w rozdziale 4.

Dodatkowo, przeprowadzono testy obliczające parametry modeli (3.65), (3.66) oraz (2.14) minimalizujące funkcje kosztu (3.69) z wykorzystaniem jednej z metod heurystycznych inspirowanych biologicznie, tj. algorytm kukułki. Wyniki przedstawiono w Dodatku C.



Rys. 3.20. Odpowiedź skokowa obiektu oraz modelu (3.65)



Rys. 3.21. Odpowiedź skokowa obiektu oraz modelu (3.66)

4. Implementacja PLC podstawowego operatora ułamkowego rzędu

Sterowniki PLC od wielu lat są siłą napędową automatyki przemysłowej. Sprzęt i oprogramowanie systemów PLC są znormalizowane (norma IEC 61131), a ich platformy programistyczne oferują potężne narzędzie do implementacji każdego algorytmu sterowania. Jednakże większość wdrożeń obejmuje sterowanie logiczne, sterowanie sekwencyjne i sterowanie PID, chociaż platformy PLC umożliwiają również realizację bardziej złożonych zadań, na przykład algorytmów sterowania opartych na specjalistycznych algorytmach lub systemów wykrywania błędów.

W tym rozdziale zostanie zaprezentowana implementacja podstawowego operatora niecałkowitego rzędu s^α na platformie PLC oraz zostaną zbadane własności czasu rzeczywistego tej implementacji. Pojedynczy operator s^α jest podstawowym składnikiem transmitancji ułamkowej $\frac{1}{Ts^\alpha+1}$. Autorka rozprawy nie bada całej transmitancji $\frac{1}{Ts^\alpha+1}$, gdyż czynnikiem decydującym o szybkości obliczeń jest czas wyznaczenia członu s^α , a także wyniki dla samego operatora s^α są uniwersalne (w sensie zbieżności i szybkości wykonania) i mogą też być wykorzystane np. do szacowania szybkości działania regulatora FOPID. Wskazane działania również są planowane do wykonania w przyszłości.

Rozważany element s^α jest aproksymowany za pomocą znanych dyskretnych przybliżeń PSE i CFE. Operator, z którym mamy do czynienia, jest istotną częścią regulatora PID ułamkowego rzędu i innych algorytmów sterowania FO. Jako przykład przedstawiono wdrożenie na platformie SIEMENS SIMATIC S7 1200 oraz SIEMENS SIMATIC S7 1500. Porównano obie zaproponowane aproksymacje PSE i CFE w sensie dokładności, zbieżności oraz czasu wykonania obliczeń operatora całkowo-różniczkowego ułamkowego rzędu na platformie PLC.

Testowany model jest dyskretną wersją modelu dla czasu ciągłego, niecałkowitego rzędu w przestrzeni stanu, zaproponowanego w rozdziale 3. Dyskretyzacja operatora całkowo-różniczkowego dokonana z wykorzystaniem omówionego wcześniej modelu została potraktowana za funkcję zależną od rzędu modelu oraz długości pamięci niezbędnej do prawidłowej estymacji operatora niecałkowitego rzędu.

Wyniki eksperymentów pokazują, że implementacja PLC elementu ułamkowego rzędu z wykorzystaniem aproksymacji PSE oraz CFE może być wykonana z wykorzystaniem podejścia obiektowego, gdzie dokładność aproksymacji jest określona przez jego rząd oraz długość pamięci. Przybliżenie CFE jest znacznie szybsze niż PSE, a jego dokładność jest porównywalna z dokładnością PSE.

4.1. Ogólne uwagi na temat implementacji specjalnych algorytmów sterowania na platformie PLC

Ogólne ramy oprogramowania PLC określa norma IEC 61131-3 (ostatnia aktualizacja została opublikowana w listopadzie 2013 r.). Standard ten został opisany w fundamentalnej książce [82], poprzednie wersje normy zostały omówione również w [83]. Wykorzystanie metod formalnych w programowaniu sterowników PLC omówiono np. w [84], przykład implementacji specjalnego, modelowego algorytmu sterowania na platformie soft PLC firmy SIEMENS przedstawiono np. w [85].

Pomimo tego, że norma IEC 61131 opisuje wszystkie aspekty programowania PLC, każdy rzeczywisty system PLC zawiera również szereg specyficznych szczegółów z nim niekompatybilnych. Należy to wziąć pod uwagę podczas każdej rzeczywistej implementacji. Przykładowo system SIEMENS 300/400/1200/1500 zapewnia bloki organizacyjne (OB), bloki danych (DB) i specyficzne typy danych PLC.

Należy tutaj zaznaczyć, że algorytm implementujący element ułamkowego rzędu ma cechy algorytmu specjalnego.

Implementacja specjalnego algorytmu sterowania w PLC powinna odbywać się z uwzględnieniem ogólnych wytycznych sformułowanych w normie. Jednak w konkretnej sytuacji nie są one wystarczające i należy również spełnić dodatkowe założenia (patrz np. [73]):

- Obliczenia należy podzielić na mniejsze, prostsze części, zaimplementowane jako osobne POU (Program Organization Unit) tj. FC (Function) oraz FB (Function Block),
- Wszystkie obliczenia niekonieczne do wyznaczenia podczas pracy w czasie rzeczywistym: wartości współczynników, samostrojania itp. należy wykonywać poza PLC. Można je wykonać za pomocą skryptów SCADA lub MATLAB, a wyniki należy przesłać do PLC. Alternatywnym pomysłem jest wyliczanie tych współczynników w PLC raz na jakiś czas (np. podczas dostrajania sterownika lub po zmianie parametrów) i zapisywanie ich do pamięci nieulotnej,

- Czas wykonania programu na platformie PLC powinien być oszacowany lub przetestowany, aby uniknąć błędów czasowych,
- Złożone wyrażenia matematyczne należy pisać przy użyciu języków tekstowych (IL, ST), implementację całego programu można wykonać przy użyciu języków graficznych (LD, FBD lub SFC),
- Algorytm musi być rozwiązywany ze stałym okresem próbkowania, którego odmierzenie musi być wykonane przez zegar czasu rzeczywistego sterownika. Szczegółowe metody zapewnienia tego zależą od konkretnego systemu. Na przykład na platformie SIEMENS taką funkcjonalność zapewniają bloki organizacyjne OB30-OB38. Są one wywoływane przez system operacyjny ze stałym przerwaniem czasowym równym h , koniecznym do zdefiniowania. Oczekuje się, że wszystkie części programu wymagane do oceny przy zachowaniu stałego czasu próbkowania, zostaną wywołane przez jeden z tych bloków. W rozważanym wypadku zastosowano blok OB30.

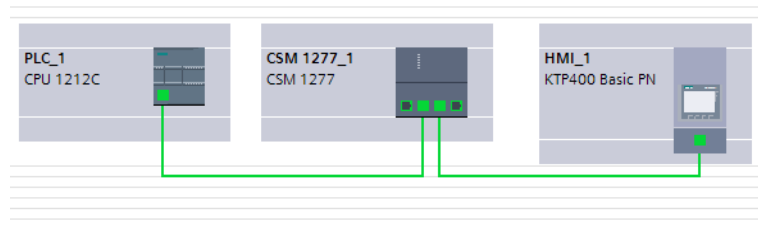
4.2. Systemy doświadczalne użyte w badaniach

W przeprowadzonych badaniach wykorzystano dwa zestawy systemów PLC ze sterownikiem PLC SIEMENS 1200 oraz PLC SIEMENS 1500. Rodzina S7 1200 cechuje się mniejszą mocą obliczeniową, niż 1500, ale jest znacznie tańsza.

4.2.1. Sterownik PLC SIEMENS 1200

Implementację operatora całkowo-różniczkowego ułamkowego rzędu wykorzystującą aproksymację PSE oraz CFE przeprowadzono na systemie PLC pokazanym na rysunku 4.1. Zestaw zawiera następujące elementy: PLC SIEMENS 1200 z CPU 1212C, panel HMI SIEMENS KTP400 oraz switch przemysłowy CSM1277. System połączony jest z komputerem PC z oprogramowaniem SIEMENS TIA PORTAL v13 poprzez PROFINET. Wszystkie parametry do eksperymentów wprowadzano za pomocą HMI, wykorzystywano go również do zbierania wyników zapisywanych na pendrive w formacie tekstowym.

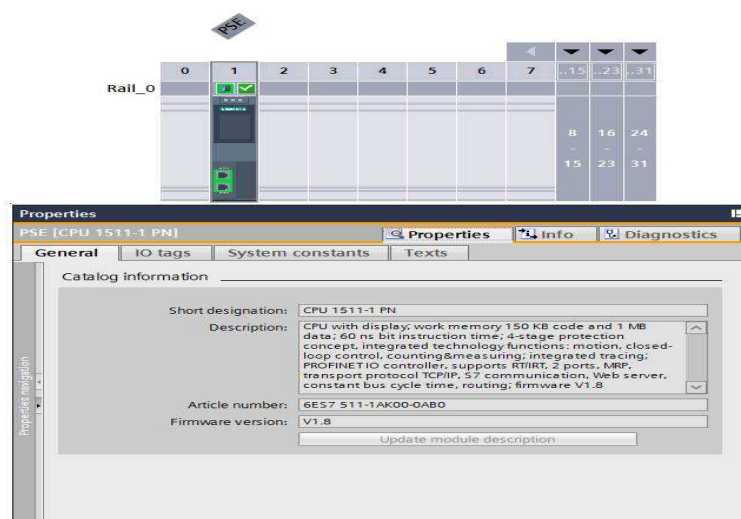
Oprogramowanie sterownika PLC SIEMENS 1200 implementujące testowane elementy FO zostało przygotowane z wykorzystaniem standardowych elementów dostępnych na platformie TIA PORTAL v13 z uwzględnieniem ogólnych uwag podanych w Rozdziale 4.1.1.



Rys. 4.1. Konstrukcja systemu doświadczalnego ze sterownikiem SIEMENS S7 1200

4.2.2. Sterownik PLC SIEMENS 1500

Testy spełnienia wymagań czasu rzeczywistego podczas realizacji obliczeń ułamkowych przeprowadzono z wykorzystaniem prostego układu PLC pokazanego na rysunku 4.2. Zawiera następujące elementy: PLC SIEMENS CPU 1511-1 PN, nr kat. 6ES7 511-1AK00-0AB0 z firmware 1.8, połączony z komputerem PC z oprogramowaniem SIEMENS TIA PORTAL v13 poprzez PROFINET. Wszystkie parametry do testów wprowadzano za pomocą aplikacji SCADA zaimplementowanej na komputerze PC z wykorzystaniem oprogramowania WinCC. Używano go również do zbierania wyników. Moduły wejścia i wyjścia nie zostały zastosowane, ponieważ nie były potrzebne do przeprowadzenia testów.



Rys. 4.2. Konstrukcja systemu doświadczalnego ze sterownikiem SIEMENS S7 1500

4.3. Implementacja operatora na platformie SIEMENS SIMATIC

Główne obszary zastosowania rachunku ułamkowego rzędu w automatyce to: sterowanie rzędem ułamkowym oraz modelowanie procesów o dynamice trudnej do opisanie innymi sposobami. Regulacja ułamkowego rzędu obejmuje głównie regulatory PID ułamkowego rzędu (FOPID). Regulatory FOPID były prezentowane przez wielu Autorów, a ich przydatność została udowodniona (zobacz [6], [86], [55], [87], [88]). Implementacja PLC sterownika FO została przedstawiona na przykład w [89].

Praktyczna implementacja regulatorów i modeli FO powoduje jednak szereg problemów, generowanych głównie przez fakt, że operator różniczkowania/całkowania ułamkowego rzędu jest niemożliwy do dokładnej implementacji i wymaga stosowania przybliżeń możliwych do cyfrowej implementacji. Można tego dokonać stosując aproksymację FOBD (Fractional Order Backward Difference), PSE (Power Series Expansion), CFE (Continuous Fraction Expansion) lub dyskretną wersję aproksymacji ORA (Ostaloup Recursive Approximation).

Rozdział ma na celu pokazanie sposobu implementacji podstawowego elementu FO s^α w PLC z uwzględnieniem podejścia obiektowego, opisanego w normie IEC61131. Rozważany element jest elementarną „cegielką” do implementacji wielu sterowników rzędu ułamkowego i modeli na platformie PLC. Do implementacji zastosowano aproksymacje PSE oraz CFE.

Oprogramowanie sterownika PLC implementujące testowane elementy FO zostało przygotowane z wykorzystaniem standardowych elementów dostępnych na platformie TIA PORTAL v13 z uwzględnieniem ogólnych uwag podanych w Rozdziale 4.1.

Główne komponenty oprogramowania dla aproksymacji PSE zostały opisane w tabeli 4.1.

Główne komponenty oprogramowania dla aproksymacji CFE zostały przedstawione w tabeli 4.2.

Tabela 4.1. Komponenty oprogramowania dla aproksymacji PSE

Jednostka Organizacyjna Programu	Typ i nr jednostki	Nazwa symboliczna	Opis
Function (FC)	FC1	PSEcalc	Funkcja obliczająca współczynniki aproksymacji PSE według (1.51).
Function Block (FB)	FB1	PSE	Blok funkcyjny obliczający odpowiedź elementu FO ze wzoru (1.55).
Organisation Block (OB)	OB1	Main	Główny blok organizacyjny programu. Zawiera warunkowe wywołanie funkcji PSEcalc. Wywołanie jest uruchamiane przyciskiem F1 na HMI, wyniki są zapisywane w nieulotnym bloku danych DB3.
Organisation Block (OB)	OB30	Cyclic interrupt	Blok organizacyjny uruchamiany przez przerwanie zegarowe ze stałym czasem próbkowania. Wywołuje blok FB1, obliczając czas odpowiedzi testowanego elementu FO.
Data Block (DB)	DB3	PSEcoeff	Blok danych zapisujący obliczone współczynniki aproksymacji PSE. Znajduje się w pamięci nieulotnej.
Data Block (DB)	DB4	PSEDB	Blok danych powiązany z blokiem funkcyjnym FB1.

Tabela 4.2. Komponenty oprogramowania dla aproksymacji CFE

Jednostka Organizacyjna Programu	Typ i nr jednostki	Nazwa symboliczna	Opis
Function (FC)	FC1	CFEcalc	Funkcja obliczająca współczynniki aproksymacji CFE według [57].
Function Block (FB)	FB1	CFE	Blok funkcyjny obliczający odpowiedź elementu FO ze wzoru (1.57).
Organization Block (OB)	OB1	Main	Główny blok organizacyjny programu. Zawiera warunkowe wywołanie funkcji CFEcalc. Wywołanie jest uruchamiane przyciskiem F1 na HMI, wyniki są zapisywane w nieulotnym bloku danych DB1.
Organization Block (OB)	OB30	Cyclic interrupt	Blok organizacyjny uruchamiany przez przerwanie zegarowe ze stałym czasem próbkowania. Wywołuje blok FB1, obliczając czas odpowiedzi testowanego elementu FO.
Data Block (DB)	DB1	CFEcoeff	Blok danych zapisujący obliczone współczynniki aproksymacji CFE. Znajduje się w pamięci nieulotnej.
Data Block (DB)	DB2	CFEDB	Blok danych powiązany z blokiem funkcyjnym FB1.

Wszystkie elementy programu zostały połączone za pomocą znaczników PLC, które są odpowiednikami zmiennych reprezentowanych bezpośrednio opisanych w normie IEC61131-3. Zmienne adresowane bezpośrednio w PLC i zmienne panelu HMI są zebrane w tabelach odpowiednio 4.3 oraz 4.4.

Tabela 4.3. Zmienne sterownika PLC SIEMENS 1200 / 1500

Name	Data Type	Logical Address	Comment
calculate_CFE	Bool	%M0.0	Conditional CFE coefficients calculating
Yout	Real	%MD4	Output signal
Uin	Real	%MD8	Input signal
M	USInt	%MB1	Order of CFE approximation
a	Real	%MD12	a coefficient
alfa	Real	%MD24	Fracional order
control_ON	Bool	%M0.1	Input signal ON
Samples_count	Int	%MW128	Number of samples

Tabela 4.4. Zmienne panelu HMI

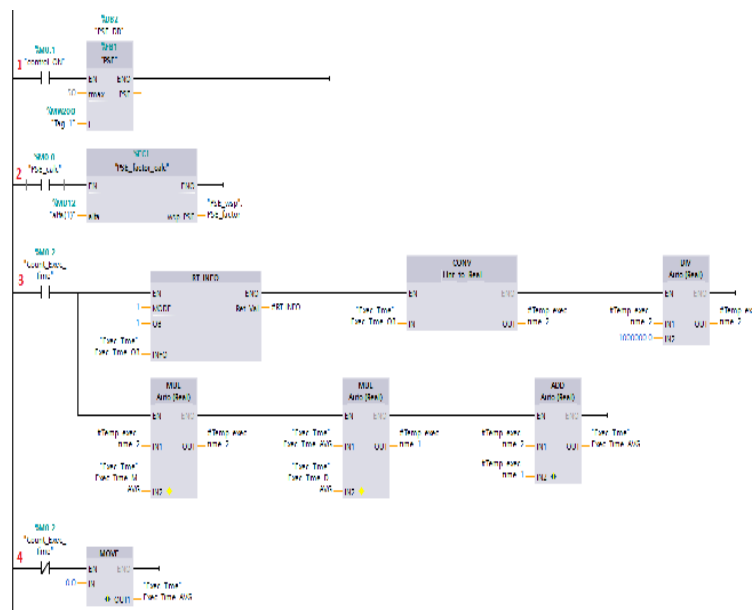
Name	Connection	PLC tag	DataType	Length	Access Method	Acquisition mode	Acquisition cycle
a	HMI_Connection_1	a	Real	4	Symbolic access	Cyclic in operation	1 s
M	HMI_Connection_1	M	USInt	1	Symbolic access	Cyclic in operation	1 s
alfa	HMI_Connection_1	alfa	Real	4	Symbolic access	Cyclic in operation	1 s
calculate_CFE	HMI_Connection_1	calculate_CFE	Bool	1	Symbolic access	Cyclic in operation	1 s
control_ON	HMI_Connection_1	control_ON	Bool	1	Symbolic access	Cyclic in operation	1 s
Yout	HMI_Connection_1	Yout	Real	4	Symbolic access	Cyclic in operation	1 s
Uin	HMI_Connection_1	Uin	Real	4	Symbolic access	Cyclic in operation	1 s
pen_removed	<No Value>	<No Value>	Bool	1	<No Value>	Cyclic in operation	1 s
sample_count	HMI_Connection_1	Samples_count	Int	2	Symbolic access	Cyclic in operation	1 s

Kody źródłowe do bloków funkcyjnych zastosowanych w implementacji programowej dla sterowników PLC zostały zamieszczone w Dodatku zawierającym kody źródłowe oprogramowania PLC. Kody źródłowe FC i FB zostały napisane w języku wysokiego poziomu STEP7 SCL, który jest odpowiednikiem języka ST z normy IEC61131-3.

4.4. Testy spełnienia wymagań czasu rzeczywistego podczas realizacji obliczeń ułamkowych

W prezentowanym rozdziale zostanie przetestowana implementacja elementarnego operatora całkowo-różniczkowego rzędu ułamkowego na platformie PLC pod kątem spełnienia wymagań czasu rzeczywistego. Rozważany element jest aproksymowany za pomocą dyskretnych przybliżeń PSE i CFE. Wdrożenie zostało wykonane na platformie SIEMENS SIMATIC S7 1500 (Rozdział 4.2.2). Zimplementowane aproksymacje PSE oraz CFE będą porównane pod kątem czasu odpowiedzi.

Testy czasów wykonania podczas obliczeń współczynników oraz oszacowania obu testowanych aproksymacji zostały wykonane za pomocą funkcji *RT_INFO* dostępnej na platformie S7 1500. Funkcje i instancje bloków funkcyjnych opisane wcześniej zostały wywołane z poziomu bloku OB30, który został skonfigurowany z użyciem języka drabinkowego. Schemat blokowy programu LD zastosowanego do testów spełnienia wymagań czasu rzeczywistego przedstawiono na rysunku 4.3.



Rys. 4.3. Diagram LD zastosowany do testów

Testy spełnienia wymagań czasu rzeczywistego podczas realizacji obliczeń ułamkowych przeprowadzono dla różnych wartości ułamkowego rzędu α oraz rzędu aproksymacji PSE $L = 10, 20, 30, 40, 50, 100, 200$. oraz rzędu aproksymacji CFE $M = 1, 2, 3, 4, 5$. Aproksymacje zastosowano z czasem próbkowania h równym 1 s, liczba pobranych próbek K_s była równa 50. Wartości dodatnie α (różniczkowanie) i ujemne (całkowanie) badano oddzielnie. Czasy wykonania dla testów aproksymacji PSE podano w tablicach 4.5 i 4.6.

Czasy wykonania dla testów aproksymacji CFE podano w tablicach 4.7 i 4.8.

Wnioski z testów spełnienia wymagań czasu rzeczywistego podczas realizacji obliczeń ułamkowych można sformułować następująco:

- Elementarny obiekt rzędu ułamkowego wyrażony transmitancją s^α można zaimplementować na platformie PLC przy użyciu znormalizowanych narzędzi programowych,
- Czas realizacji implementacji CFE jest znacznie krótszy niż w przypadku PSE,

Tabela 4.5. Czasy wykonywania funkcji PSEcoeff (obliczanie współczynników d_l dla aproksymacji PSE) w [ms]

	L=10	L=20	L=30	L=40	L=50	L=100	L=200
$\alpha=0.25$	0,9	0,9	0,9	0,9	1	0,9	1
$\alpha=0.5$	0,9	0,9	0,9	0,9	0,9	1	1
$\alpha=0.75$	0,9	1	1	0,9	1	1	1
$\alpha=-0.25$	0,9	0,9	0,9	1	0,9	1	1
$\alpha=-0.5$	0,9	0,9	1	1	1	1	1
$\alpha=-0.75$	0,9	0,9	1	1	1	1	1

Tabela 4.6. Czasy wykonywania bloku funkcyjnego instancji PSE (obliczenie aproksymacji PSE) w OB30 w [ms]

	L=10	L=20	L=30	L=40	L=50	L=100	L=200
$\alpha=0.25$	3,7	6,2	8,0	9,1	9,5	36,3	143,2
$\alpha=0.5$	3,8	6,3	8,1	9,1	9,6	36,5	143,4
$\alpha=0.75$	3,8	6,3	8,2	9,2	9,5	36,5	143,4
$\alpha=-0.25$	3,7	6,3	8,1	9,1	9,5	36,4	143,3
$\alpha=-0.5$	3,8	6,4	8,2	9,2	9,6	36,5	143,4
$\alpha=-0.75$	3,8	6,3	8,1	9,3	9,5	36,6	143,5

Tabela 4.7. Czasy wykonywania funkcji CFEEcoeff (obliczenie współczynników w_m i v_m dla aproksymacji CFE) w [ms]

	M=1	M=2	M=3	M=4	M=5
$\alpha=0.25$	0,61	0,63	0,67	0,69	0,77
$\alpha=0.5$	0,61	0,63	0,67	0,69	0,76
$\alpha=0.75$	0,62	0,63	0,68	0,69	0,76
$\alpha=-0.25$	0,63	0,64	0,68	0,69	0,77
$\alpha=-0.5$	0,63	0,65	0,68	0,7	0,78
$\alpha=-0.75$	0,62	0,63	0,67	0,68	0,77

Tabela 4.8. Czasy wykonywania instancji bloku funkcyjnego CFE (obliczanie aproksymacji CFE) w OB30 w [ms]

	M=1	M=2	M=3	M=4	M=5
$\alpha=0.25$	0,15	0,15	0,15	0,14	0,14
$\alpha=0.5$	0,15	0,14	0,14	0,14	0,14
$\alpha=0.75$	0,15	0,14	0,14	0,14	0,14
$\alpha=-0.25$	0,14	0,14	0,14	0,14	0,14
$\alpha=-0.5$	0,15	0,14	0,14	0,14	0,14
$\alpha=-0.75$	0,15	0,14	0,14	0,14	0,14

- Czasy podczas wykonywania obliczeń FO silnie zależą od rzędu aproksymacji. Implikuje to fakt, że aproksymacja CFE jest znacznie szybsza niż PSE, która wymaga znacznie większej długości pamięci niezbędnej do uzyskania dobrej dokładności.

Wyniki eksperymentów pokazują, że implementacja PLC elementu ułamkowego rzędu dla aproksymacji CFE jest znacznie szybsze niż dla aproksymacji PSE. Zastosowanie aproksymacji CFE pozwala na otrzymanie poprawnie działającego elementu rzędu ułamkowego o rozsądnym rzędzie i krótkim czasie wykonania. Można to wskazać jako zaletę tej metody w porównaniu z aproksymacją PSE, gdzie podobna dokładność wymaga od nas zastosowania znacznie wyższego rzędu i bardziej złożonego modelu. Pozwala to na sformułowanie ostatecznego wniosku, że metoda CFE może być rekomendowana do stosowania w trudnych implementacjach czasu rzeczywistego.

4.5. Testy dokładności i zbieżności

W rozdziale zostaną przedstawione wyniki testów dotyczących badania dokładności oraz zbieżności aproksymacji PSE oraz CFE.

Dokładność obu rozważanych aproksymacji PSE (1.55) oraz CFE (1.57) zostanie oszacowana za pomocą standardowej funkcji kosztu MSE (Medium Square Error):

$$MSE = \frac{1}{K_s} \sum_{k=1}^{K_s} \left(y(kh) - y_{PSE/CFE}^+(k) \right)^2 \quad (4.1)$$

gdzie K_s to liczba wszystkich próbek sygnału, y to analityczna odpowiedź czasowa obliczona w dyskretnych krokach czasowych kh , $y_{PSE/CFE}^+$ to odpowiedź czasowa aproksymacji PSE/CFE, obliczane na tej samej siatce czasowej i w odniesieniu do (1.55) lub (1.57). Jeśli założymy, że sygnał wejściowy $u(t)$ jest funkcją Heaviside'a: $u(t) = 1(t)$, to $y(t) = y_{an}(t)$, gdzie y_{an} jest

analityczną odpowiedzią skokową elementu o transmitancji s^α , opisaną następująco:

$$y_{an}(t) = \frac{t^{-\alpha}}{\Gamma(1 - \alpha)}. \quad (4.2)$$

Analityczny wzór na odpowiedź skokową (4.2) będzie interpretowany jako wzorzec do oszacowania dokładności obu testowanych implementacji PLC. Metodą wykorzystaną do wyboru elementu optymalnego (wartości parametrów poszczególnych modeli) będzie algorytm deterministyczny Nelder-Meada realizowany przy użyciu funkcji Matlaba *fminsearch*.

Pokazane zostaną również wykresy zależności wartości funkcji kosztu MSE i czasu trwania obliczeń od długości pamięci L w aproksymacji PSE oraz wartości rzędu M w aproksymacji CFE dla różnych rzędów α .

Testy modelu dla aproksymacji PSE oraz CFE przeprowadzono z wykorzystaniem systemu eksperymentalnego PLC SIEMENS 1200, który został pokazany w rozdziale 4.2.1.

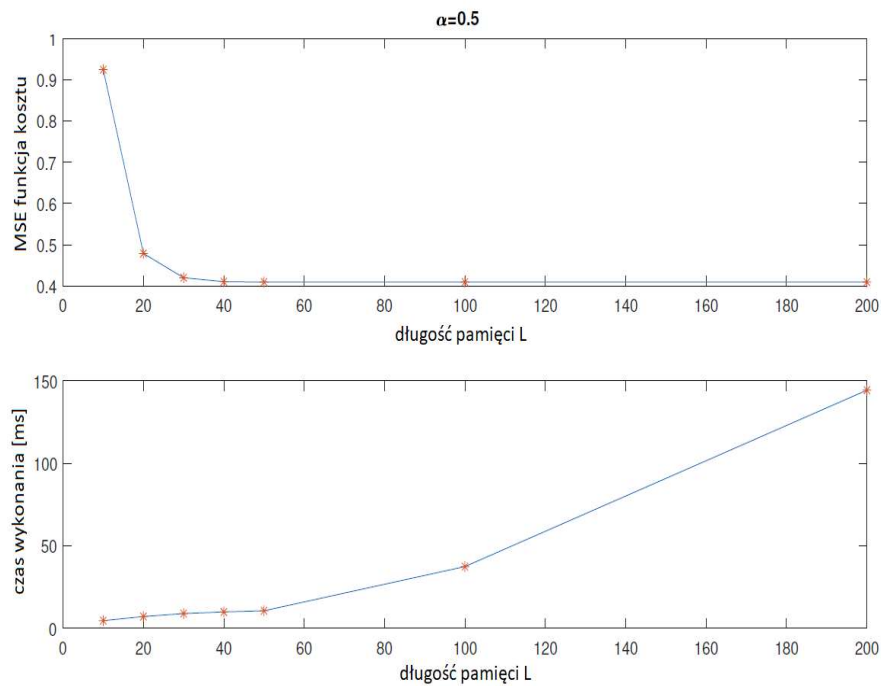
Najpierw przedstawione zostaną własności aproksymacji PSE. Jej dokładność została sprawdzona za pomocą funkcji kosztu MSE (4.1). Testy przeprowadzono dla różnych wartości ułamkowego rzędu α oraz rzędu aproksymacji PSE $L = 10, 20, 30, 40, 50, 100$ i 200 . Zastosowano aproksymację PSE przy czasie próbkowania h równym 1 s, liczba pobranych próbek K_s była równa 50 . Wartości dodatnie α (różniczkowanie) i ujemne (całkowanie) badano oddzielnie. Wartości funkcji kosztu (4.1) dla wszystkich testów podano w tablicach 4.9 i 4.10. Wykresy zależności funkcji kosztu MSE i czasu odpowiedzi od wartości L przedstawiono na rysunkach 4.4 i 4.5.

Tabela 4.9. Funkcja kosztu MSE (4.1) dla aproksymacji PSE dla dodatnich wartości α

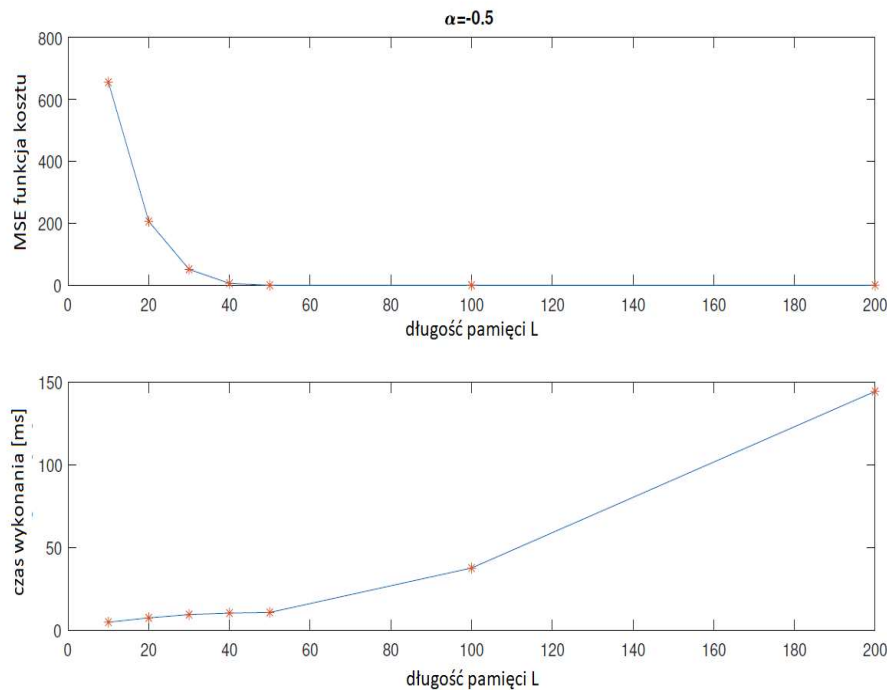
	$\alpha = 0.25$	$\alpha = 0.50$	$\alpha = 0.75$
$L=10$	1.1664	0.9246	1.1387
$L=20$	0.2680	0.4788	1.0750
$L=30$	0.1145	0.4201	1.0686
$L=40$	0.0850	0.4104	1.0678
$L=50$	0.0820	0.4095	1.0677
$L=100$	0.0820	0.4095	1.0677
$L=200$	0.0820	0.4095	1.0677

Tabela 4.10. Funkcja kosztu MSE (4.1) dla aproksymacji PSE dla ujemnych wartości α

	$\alpha = -0.25$	$\alpha = -0.50$	$\alpha = -0.75$
$L=10$	35.4139	655.6779	6265.6482
$L=20$	9.3249	206.3641	2318.2036
$L=30$	2.1109	51.3412	638.5235
$L=40$	0.2757	6.1833	80.5312
$L=50$	0.0501	0.1685	0.2527
$L=100$	0.0501	0.1685	0.2527
$L=200$	0.0501	0.1685	0.2527



Rys. 4.4. Zależność funkcji kosztu MSE i czasu trwania od długości pamięci L dla $\alpha = 0,5$ w aproksymacji PSE



Rys. 4.5. Zależność funkcji kosztu MSE i czasu trwania od długości pamięci L dla $\alpha = -0,5$ w aproksymacji PSE

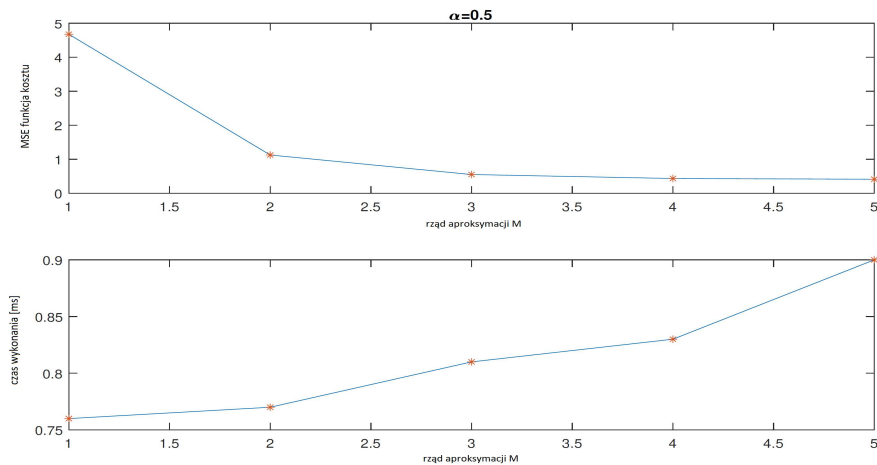
Następnie przetestowano aproksymację CFE. Jej dokładność również oszacowano za pomocą funkcji kosztu MSE (4.1). Testy przeprowadzono dla różnych wartości rzędu ułamkowego α i różnych wartości rzędu aproksymacji CFE równych $M = 1 \dots 5$. Zastosowano aproksymację Tustina ($a = 1$) przy czasie próbkowania h równym 1 s, liczba pobranych próbek K_s była równa 50. Dodatnie wartości α (różniczkowanie) i ujemne wartości (całkowanie) testowano oddzielnie. Wartości funkcji kosztu (4.1) dla wszystkich testów podane są w tablicach 4.11 i 4.12. Wykresy zależności funkcji kosztu MSE i czasu działania w zależności od długości M przedstawiono na rysunkach 4.6 i 4.7.

Tabela 4.11. Funkcja kosztu MSE (4.1) dla aproksymacji CFE dla dodatnich wartości α

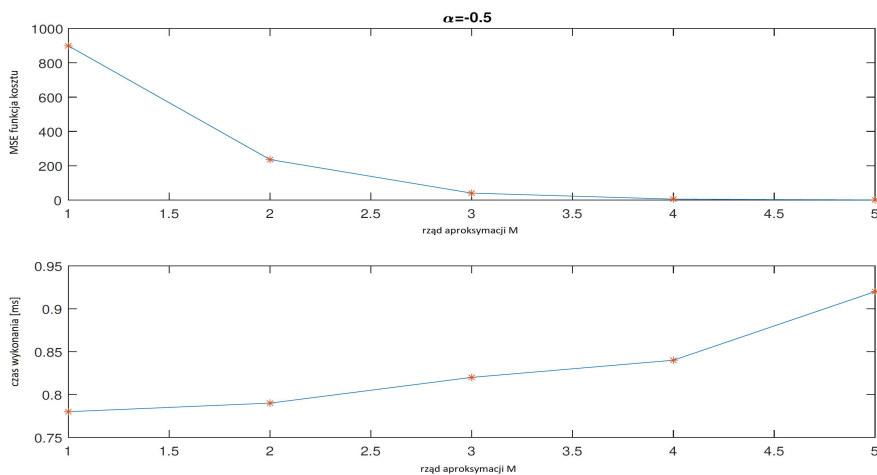
	$\alpha = 0.25$	$\alpha = 0.50$	$\alpha = 0.75$
$M=1$	5.0838	4.6758	2.2865
$M=2$	1.1845	1.1265	1.2097
$M=3$	0.3229	0.5522	1.0920
$M=4$	0.1256	0.4357	1.0720
$M=5$	0.0885	0.4136	1.0690

Tabela 4.12. Funkcja kosztu MSE (4.1) dla aproksymacji CFE dla ujemnych wartości α

	$\alpha = -0.25$	$\alpha = -0.50$	$\alpha = -0.75$
$M=1$	69.0932	899.0891	5281.0921
$M=2$	19.7334	235.8201	906.2536
$M=3$	4.1168	40.2426	109.7010
$M=4$	0.6651	5.3124	11.6897
$M=5$	0.1263	0.7404	1.4017

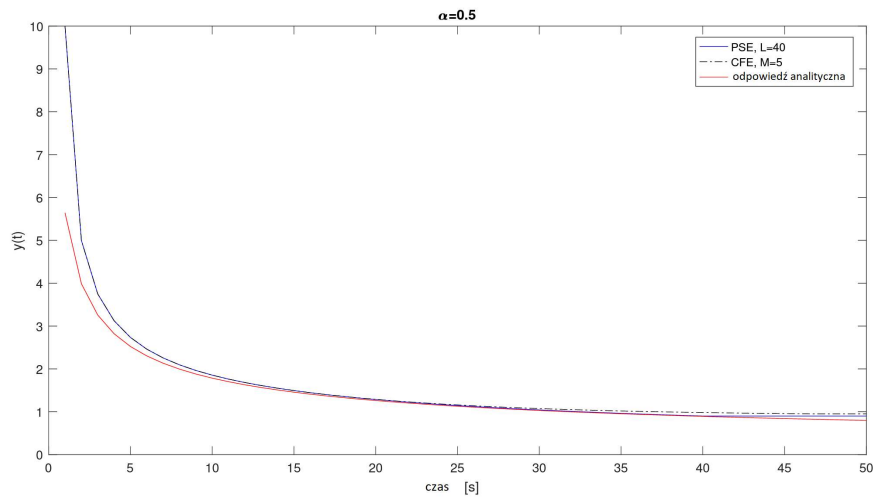


Rys. 4.6. Zależność funkcji kosztu MSE i czasu trwania od długości pamięci L dla $\alpha = 0,5$ w aproksymacji CFE

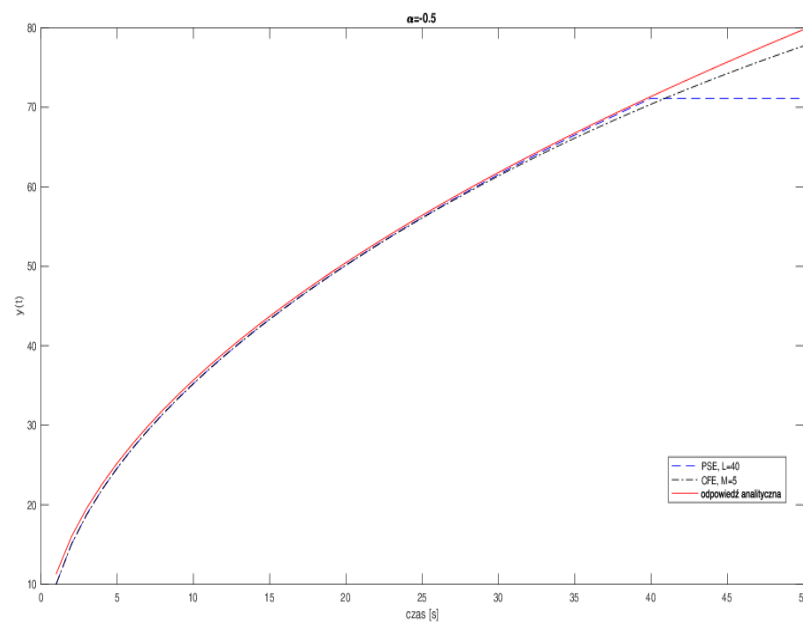


Rys. 4.7. Zależność funkcji kosztu MSE i czasu trwania od długości pamięci L dla $\alpha = -0,5$ w aproksymacji PSE

Odpowiedzi skokowe aproksymacji PSE w porównaniu z CFE oraz wzorem analitycznym (4.2) podano na rysunkach 4.8 i 4.9.



Rys. 4.8. Odpowiedzi skokowe aproksymacji PSE i CFE w porównaniu z odpowiedzią analityczną dla $\alpha = 0,5$



Rys. 4.9. Odpowiedzi skokowe aproksymacji PSE i CFE w porównaniu z odpowiedzią analityczną dla $\alpha = -0,5$

Wykresy odpowiedzi skokowej obiektu oraz modelu z różnymi wartościami rzędu α oraz różnymi wartościami długości pamięci L dla aproksymacji PSE oraz różnymi rzędami M dla aproksymacji CFE przedstawiono odpowiednio w tabelach 4.13 oraz 4.14.

Tabela 4.13. Odpowiedź skokowa: obiektu (czerwona linia) oraz modelu z różnymi wartościami rzędu α oraz różnymi rzędami L dla aproksymacji PSE

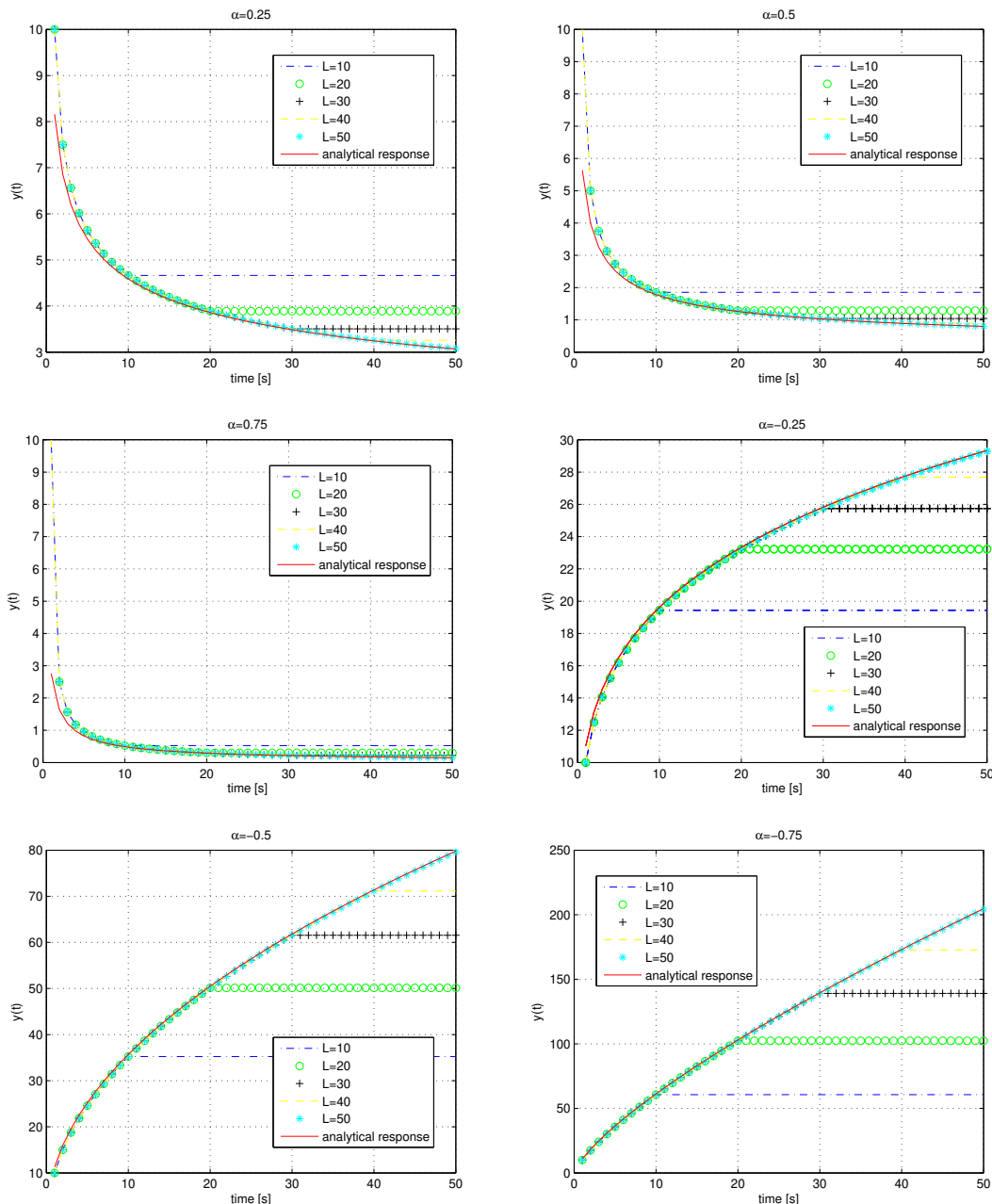
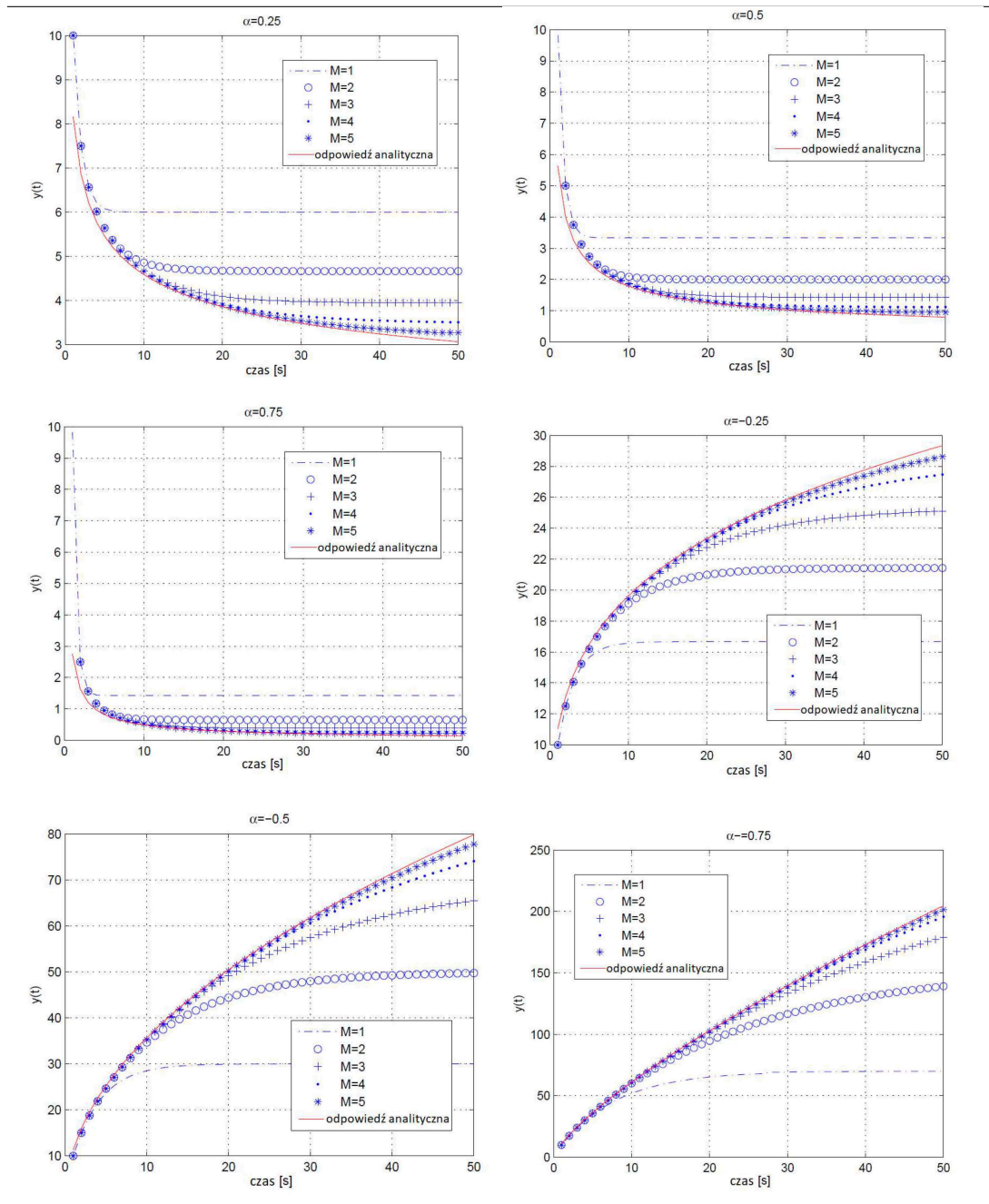


Tabela 4.14. Odpowiedź skokowa: obiektu (czerwona linia) oraz modelu z różnymi wartościami rzędu α oraz różnymi rzędami M dla aproksymacji CFE



Wnioski z badań przeprowadzonych w tym rozdziale można sformułować następująco:

- Elementarny układ rzędu ułamkowego, wyrażony przez transmitancję s^α , można zaimplementować na platformie PLC przy użyciu znormalizowanych narzędzi programowych i podejścia obiektowego zalecanego przez normę IEC 61131,
- Dokładność modelu jest określona przez rząd aproksymacji: wyższy rząd daje większą dokładność,
- Aproksymacja CFE wymaga od nas użycia znacznie krótszej pamięci niż metoda PSE,
- Czas realizacji implementacji CFE jest znacznie krótszy niż w przypadku PSE. Jednocześnie dokładność obu prezentowanych metod jest porównywalna,
- Zastosowanie aproksymacji CFE pozwala uzyskać poprawnie działający element rzędu ułamkowego o sensownym rzędzie. Można to wskazać na zaletę tej metody w przeciwieństwie do aproksymacji PSE, gdzie podobna dokładność wymaga od nas użycia znacznie wyższego rzędu i bardziej złożonego modelu.

Wyniki eksperymentów pokazują, że implementacja PLC elementu ułamkowego rzędu dla aproksymacji CFE jest znacznie szybsza niż dla aproksymacji PSE. Jej dokładność jest akceptowalnie mniejsza. Pozwala to na sformułowanie ostatecznego wniosku, że metoda CFE może być rekomendowana do stosowania w trudnych implementacjach czasu rzeczywistego.

Podsumowanie

Niniejsza praca zawiera opracowanie, analizę teoretyczną, identyfikację oraz weryfikację doświadczalną dla nowych, zaproponowanych przez autorkę modeli niecałkowitego rzędu dla procesów przewodnictwa cieplnego.

Porównano cztery modele ciągłe zdefiniowane w przestrzeni stanu: pierwszy klasyczny, rzędu całkowitego oraz trzy nowe modele niecałkowitego rzędu. Wszystkie proponowane modele niecałkowitego rzędu są dokładniejsze w sensie funkcji kosztu MSE niż ich odpowiedniki całkowitego rzędu. Najlepszym modelem pod względem minimalizacji funkcji kosztu MSE jest model o niecałkowitym rzędzie zarówno wzdłuż długości, jak i czasu.

Przedstawiono nową, efektywną obliczeniowo dyskretną metodę rozwiązywania równań niecałkowitego rzędu, bazującą na aproksymacji CFE. Zaproponowano również dyskretne modele w przestrzeni stanu wykorzystujące aproksymacje dyskretne FOBD oraz CFE. Opracowano i zweryfikowano doświadczalnie nowe modele transmitancyjne niecałkowitego rzędu w postaci hybrydowych transmitancji ciągłych i dyskretnych.

W ostatniej części dokonano implementacji na sterowniku PLC podstawowego elementu ułamkowego rzędu z użyciem dyskretnych aproksymacji FOBD i CFE. Wyniki doświadczalne wskazują, że model dyskretny może być implementowany na platformach cyfrowych PLC.

Na podstawie przeprowadzonych badań oraz uzyskanych wyników można stwierdzić, że tezy pracy zostały udowodnione, gdyż:

1. Modele ciągłe w przestrzeni stanu zostały opracowane i zweryfikowane doświadczalnie.
2. Nowe hybrydowe modele transmitancyjne zostały opracowane i zweryfikowane doświadczalnie.
3. Modele dyskretne w przestrzeni stanu zostały opracowane i zweryfikowane doświadczalnie. Badania testowe dyskretnych implementacji PLC podstawowego operatora ułamkowego wykazały, że może on być zaimplementowany na platformie PLC ze spełnieniem wymagań dotyczących zarówno dokładności, jak i wymagań czasu rzeczywistego.

Ze względu na brak czasu nie udało się bezpośrednio zaimplementować żadnego z modeli na PLC i jest to w planach dalszych badań. Jako dalsze prace w tym kierunku można proponować układy predykcji i detekcji awarii związanych z przegrzewaniem się elementów oraz konstrukcję algorytmów sterowania bazujących na modelu obiektu wykorzystujących zbudowane modele. Innym kierunkiem dalszych badań jest uwzględnienie niepewności parametrów modeli. Niepewność ta może być opisana np. przez liczby przedziałowe. W najbliższej przyszłości planuje się również optymalizację proponowanych modeli niecałkowitego rzędu z użyciem metod metaheurystycznych inspirowanych przez naturę.

Dodatek A: Kody źródłowe oprogramowania w Matlabie

```
1 clc
2 clear
3 tic
4
5 for i= 1:1:30
6 wej=[1.5145,1.4176, 39.6257, 71.0073];
7
8 % dla rzędu = 1
9 wej=[1.5433,99.1564];
10 [wynik(i,:)]=fminsearch(@(wej) piecyk_dyskr_tf_del_1rz(wej,i),
    wej);
11 jak(i)= piecyk_dyskr_tf_del_1rz(wynik(i,:),i);
12 i
13
14 % dla rzędu = 2
15 %[wynik(i,:)]=fminsearch(@(wej) piecyk_dyskr_tf_del_2rz(wej,i),
    wej);
16 %jak(i)= piecyk_dyskr_tf_del_2rz(wynik(i,:),i);
17 %i
18
19 % dla rzędu - CAŁKOWITEGO 2
20 %wej=[39.6257, 71.0073];
21 %[wynik(i,:)]=fminsearch(@(wej) piecyk_dyskr_tf_del_2rz_CALK(
    wej,i),wej);
22 %jak(i)= piecyk_dyskr_tf_del_2rz_CALK(wynik(i,:),i);
23 %i
```

```

24
25 % dla rzędu - CAŁKOWITEGO 1
26 %wej=[99.1564];
27 %[wynik(i,:)]=fminsearch(@(wej) piecyk_dyskr_tf_del_1rz_CALC(
    wej,i),wej);
28 %jak(i)= piecyk_dyskr_tf_del_1rz_CALC(wynik(i,:),i);
29 %i
30
31 end
32 toc
33 jak'
34
35 plot(jak)
36 wynik

```

```

1 function [wsp_w,wsp_v]=dfod1_EG(order,Ts,a,alfa)
2
3 %współczynniki aproksymacji CFE:
4
5 if order==5
6     wsp_w=[1, -alfa, 2*alfa^2/5, -alfa/5-2*alfa^3/35, alfa^2/5,
7         -alfa/5];
8     wsp_v=[1, alfa, 2*alfa^2/5, alfa/5+2*alfa^3/35, alfa^2/5,
9         alfa/5];
10
11 else if order==3
12     wsp_w=[1, -alfa, alfa^2/3, -alfa/3];
13     wsp_v=[1, alfa, alfa^2/3, alfa/3];
14
15 else
16     wsp_w=[];
17     wsp_v=[];
18
19 end
20
21 end

```

```

1 %Model dyskretny hybrydowy z opóźnieniem i jednym rzędem
2     całkowitym

```

```
3 function [MSE]=piecyk_dyskr_tf_del_lrz_CALC(wej,nr)
4 %Ekstrakcja parametrów wejściowych:
5 alfa=1;
6 Talfa=wej(1);
7
8 %Wymiary:
9 M=5; %rzad aproksymacji dyskretnej CFE
10 N=nr;
11 %N=16; %wielkosc opoznienia w ilosci okresow probkowania
   OPTYMALNE dla 1 rzędu: N=16
12 %okres probkowania - stały przy pomiarach z modelu:
13 h=1;
14 Tfin=300; %czas finalny:
15 %czesc opozniajaca:
16 mD=zeros(1,N+1);
17 mD(1)=1;
18 sysD=tf([1],mD,h);
19 %aproksymacja czynnika s^alfa:
20 %[lfoda,mfoda]=alaouiCFE(M,h,1,alfa);
21 [lfoda,mfoda]=dfod1_EG(M,h,1,alfa);
22 % cała czesc ułamkowa alfa jako licznik i mianownik:
23 lfa=mfoda;
24 mfa=Talfa*lfoda+mfoda;
25 sysF=tf(lfa,mfa,h);
26
27 %całosc: opoznienie+ułamki
28 sysW=series(sysD,sysF);
29 %Załadowanie wyników pomiarów:
30 load piecyk_3_cels;
31 sensor_no=1; %nr czujnika
32 y=piecyk_3_cels(1:300,sensor_no);
33
34 czas=1:1:length(y);
35 ym=step(sysW,czas);
36
37 %Skalowanie:
```

```

38 k=(y(300)-y(1))/ym(300);
39 %funkcja kosztu:
40 e2=( (y-y(1))-k*ym).^2;
41 MSE=sum(e2)/300;
42 %wykres kontrolny:
43 plot(czas,y(1)+k*ym,'blue',czas,y(1)+(y-y(1)),'red');
44 %plot(czas,y(1),'blue');
45 grid;
46 xlabel('time [s]')
47 ylabel('Temperature [^oC]')

```

```

1 %Model dyskretny hybrydowy z opóźnieniem i jednym rzędem
   niecałkowitymi
2
3 function [MSE]=piecyk_dyskr_tf_del_lrz(wej,nr)
4 %Ekstrakcja parametrów wejściowych:
5 alfa=wej(1);
6 Talfa=wej(1);
7
8 %Wymiary:
9 M=5; %rzad aproksymacji dyskretnej CFE
10 N=nr;
11 %N=16; %wielkosc opoznienia w ilosci okresow probkowania
   OPTYMALNE dla 1 rzędu: N=16
12 %okres probkowania - stały przy pomiarach z modelu:
13 h=1;
14 Tfin=300; %czas finalny:
15 %czesc opozniajaca:
16 mD=zeros(1,N+1);
17 mD(1)=1;
18 sysD=tf([1],mD,h);
19 %aproksymacja czynnika s^alfa:
20 %[lfoda,mfoda]=alaouiCFE(M,h,1,alfa);
21 [lfoda,mfoda]=dfodl_EG(M,h,1,alfa);
22 % cała czesc ułamkowa alfa jako licznik i mianownik:
23 lfa=mfoda;

```

```

24 mfa=Talfa*lfoda+mfoda;
25 sysF=tf(lfa,mfa,h);
26
27 %całosc: opoznienie+ułamki
28 sysW=series(sysD,sysF);
29 %Załadowanie wyników pomiarów:
30 load piecyk_3_cels;
31 sensor_no=1; %nr czujnika
32 y=piecyk_3_cels(1:300,sensor_no);
33
34 czas=1:1:length(y);
35 ym=step(sysW,czas);
36
37 %Skalowanie:
38 k=(y(300)-y(1))/ym(300);
39 %funkcja kosztu:
40 e2=(y-y(1)-k*ym).^2;
41 MSE=sum(e2)/300;
42 %wykres kontrolny:
43 plot(czas,y(1)+k*ym,'blue',czas,y(1)+(y-y(1)),'red');
44 %plot(czas,y(1),'blue');
45 grid;
46 xlabel('time [s]')
47 ylabel('Temperature [°C]')

```

```

1 %Model dyskretny hybrydowy z opoznieniem i dwoma rzędami
   całkowanym
2
3 function [MSE]=piecyk_dyskr_tf_del_2rz_CALC(wej,nr)
4 %Ekstrakcja parametrów wejściowych
5 alfa=1;
6 beta=1;
7 Talfa=wej(1);
8 Tbeta=wej(2);
9
10 %Wymiary:

```

```

11 M=5; %rzad aproksymacji dyskretnej CFE
12 N=nr; %wielkosc opoznienia w ilosci okresow probkowania
13 %okres probkowania - stały przy pomiarach z modelu:
14 h=1;
15 Tfin=300; %czas finalny
16 %czesc opozniajaca:
17 mD=zeros(1,N+1);
18 mD(1)=1;
19 sysD=tf([1],mD,h);
20
21 %aproksymacja czynnika s^alfa:
22 %[lfoda,mfoda]=alaouiCFE(M,h,1,alfa);
23 [lfoda,mfoda]=dfod1_EG(M,h,1,alfa);
24 % cała czesc ułamkowa alfa jako licznik i mianownik:
25 lfa=mfoda;
26 mfa=Talfa*lfoda+mfoda;
27 sysF=tf(lfa,mfa,h);
28 %aproksymacja czynnika s^beta:
29 [lfodb,mfodb]=dfod1_EG(M,h,1,alfa);
30 %cała czesc ułamkowa beta jako licznik i mianownik:
31 lfb=mfodb;
32 mfb=Tbeta*lfodb+mfodb;
33 sysFb=tf(lfb,mfb,h);
34 %cała czesc ułamkowa
35 sysF=series(sysFa,sysFb);
36 %calosc: opoznienie+ulamki
37 sysW=series(sysD,sysF);
38 %Załadowanie wyników pomiarow:
39 load piecyk_3_cels;
40 sensor_no=1; %numer czunika
41 y=piecyk_3_cels(1:300,sensor_no);
42
43 czas=1:1:length(y);
44 ym=step(sysW,czas);
45
46 %Skalowanie:

```

```

47 k=(y(300)-y(1))/ym(300);
48 %funkcja kosztu:
49 e2=(y-y(1)-k*ym).^2;
50 MSE=sum(e2)/300;
51 %wykres kontrolny:
52 plot(czas,y(1)+k*ym,'blue',czas,y(1)+(y-y(1)),'red');
53 %plot(czas,y(1),'blue');
54 grid;
55 xlabel('time [s]')
56 ylabel('Temperature [^oC]')

```

```

1  %Model dyskretny hybrydowy z opóźnieniem i dwoma rzędami
   niecałkowitym
2
3  function [MSE]=piecyk_dyskr_tf_del_2rz(wej,nr)
4  %Ekstrakcja parametrów wejściowych
5  alfa=wej(1);
6  beta=wej(2);
7  Talfa=wej(1);
8  Tbeta=wej(2);
9
10 %Wymiary:
11 M=5; %rzęd aproksymacji dyskretniej CFE
12 N=nr; %wielkość opóźnienia w ilości okresów próbkowania
13 %okres próbkowania - stały przy pomiarach z modelu:
14 h=1;
15 Tfin=300; %czas finalny
16 %część opóźniająca:
17 mD=zeros(1,N+1);
18 mD(1)=1;
19 sysD=tf([1],mD,h);
20
21 %aproksymacja czynnika s^alfa:
22 % [lfoda,mfoda]=alaouiCFE(M,h,1,alfa);
23 [lfoda,mfoda]=dfod1_EG(M,h,1,alfa);
24 % cała część ułamkowa alfa jako licznik i mianownik:

```

```

25 lfa=mfoda;
26 mfa=Talfa*lfoda+mfoda;
27 sysF=tf(lfa,mfa,h);
28 %aproxymacja czynnika s^beta:
29 [lfodb,mfodb]=dfod1_EG(M,h,1,alfa);
30 %cała czesc ułamkowa beta jako licznik i mianownik:
31 lfb=mfodb;
32 mfb=Tbeta*lfodb+mfodb;
33 sysFb=tf(lfb,mfb,h);
34 %cała czesc ułamkowa
35 sysF=series(sysFa,sysFb);
36 %calosc: opoznienie+ulamki
37 sysW=series(sysD,sysF);
38 %Załadowanie wyników pomiarow:
39 load piecyk_3_cels;
40 sensor_no=1; %numer czunika
41 y=piecyk_3_cels(1:300,sensor_no);
42
43 czas=1:1:length(y);
44 ym=step(sysW,czas);
45
46 %Skalowanie:
47 k=(y(300)-y(1))/ym(300);
48 %funkcja kosztu:
49 e2=((y-y(1))-k*ym).^2;
50 MSE=sum(e2)/300;
51 %wykres kontrolny:
52 plot(czas,y(1)+k*ym,'blue',czas,y(1)+(y-y(1)),'red');
53 %plot(czas,y(1),'blue');
54 grid;
55 xlabel('time [s]')
56 ylabel('Temperature [^oC]')

```


Dodatek B: Kody źródłowe oprogramowania PLC

test_FO_s1200 / PLC_1 [CPU 1212C AC/DC/Rly] / Program blocks

CFE_calc [FC1]

CFE_calc Properties

General

Name	CFE_calc	Number	1	Type	FC	Language	SCL
Numbering	automatic						

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

CFE_calc

Name	Data type	Default value	Comment
▼ Input			
n	USInt		rzad aproksymacji
T	Real		okres probkowania do obliczen
a	Real		współczynnik
r	Real		rzad ulamkowy do zamodelowania
▼ Output			
▼ CFE_licz	Array[0..5] of Real		
CFE_licz[0]	Real		
CFE_licz[1]	Real		
CFE_licz[2]	Real		
CFE_licz[3]	Real		
CFE_licz[4]	Real		
CFE_licz[5]	Real		
▼ CFE_mian	Array[0..5] of Real		
CFE_mian[0]	Real		
CFE_mian[1]	Real		
CFE_mian[2]	Real		
CFE_mian[3]	Real		
CFE_mian[4]	Real		
CFE_mian[5]	Real		
InOut			
▼ Temp			
k	Real		współczynnik
b0	Real		
b1	Real		
b2	Real		
b3	Real		
b4	Real		
b5	Real		
a0	Real		
a1	Real		
a2	Real		
a3	Real		
a4	Real		
a5	Real		
m	Int		wskaznik petli do inicjalizacji
Constant			
▼ Return			
CFE_calc	Void		

```

0001
0002
0003 //Funkcja różniczkowania i całkowania ułamkowego ze strony
0004 //Petrasa:
0005 //
0006 //http://people.tuke.sk/igor.podlubny/USU/matlab/petras/dfod1.m
0007 //function sysdfod=dfod1(n,T,a,r)
0008 // sysdfod=dfod1(n,T,a,r): new digital fractional order differentiator
0009 // AND integrator
0010 //
0011 // Output: :=>
0012 // Discrete system in the form OF the rational approximation - IIR filter
0013 // obtained BY continued fraction expansion OF a new generating function.
0014 //
0015 // Inputs: <:=
0016 // n: order OF truncation n:=(1 2 3 4 5)
0017 //
0018 //T: sampling period in [sec]
0019 // a: weighting factor between Euler AND Tustin rules ( 0 <:= a <:= 1 )
0020 // 0 - Euler rule, 1 - Tustin rule, 1/7 - Al-Alaoui rule, etc.
0021 // r: approximated fractional order (s**r), r is generally real number
0022 //
0023 // Author: Dr. Ivo Petras (ivo.petras@mail.com)
0024 // URL: http://ivopetras.tripod.com/
0025 //
0026 // Note: This approach is based on the original Al-Alaoui's work.
0027 //
0028 // Copyright (c), 2003.
0029 //

```

Totally Integrated Automation Portal		
<pre> 0030 //inicjalizacja: 0031 FOR #m := 0 TO 5 DO 0032 #CFE_licz[#m] := 0.0; 0033 #CFE_mian[#m] := 0.0; 0034 END_FOR; 0035 0036 0037 0038 #k := ((1+#a)/#T)**#r; 0039 0040 IF (#n=1) THEN 0041 #b0 := 2 / (#a * #r + #a + #r - 1); 0042 #b1 := - (#r + #a * #r - #a + 1) / (#a * #r + #a + #r - 1); 0043 #a0 := 2 / (#a * #r + #a + #r - 1); 0044 #a1 := 1; 0045 //przypisanie wyniku: 0046 #CFE_licz[0] := #b0; 0047 #CFE_licz[1] := #b1; 0048 // 0049 #CFE_mian[0] := #a0; 0050 #CFE_mian[1] := #a1; 0051 //sys1:=tf(k*[b0 b1],[a0 a1],T,'Variable', 'z**(-1)'); 0052 //sysdfod:=sys1; 0053 ELSIF (#n=2) THEN 0054 #b0 := 12 / (#a ** 2 * #r ** 2 + 3 * #a ** 2 * #r + 2 * #a * #r ** 2 + 2 * #a ** 2 + #r ** 2 - 8 * #a - 3 * #r + 2); 0055 #b1 := (12 * #a - 6 * #r - 6 * #a * #r - 12) / (#a ** 2 * #r ** 2 + 3 * #a ** 2 * #r + 2 * #a * #r ** 2 + 2 * #a ** 2 + #r ** 2 - 8 * #a - 3 * #r + 2); 0056 #b2 := (#a ** 2 * #r ** 2 - 3 * #a ** 2 * #r + 2 * #a * #r ** 2 + #r ** 2 - 8 * #a + 3 * #r + 2 * #a ** 2 + 2) / (#a ** 2 * #r ** 2 + 3 * #a ** 2 * #r + 2 * #a * #r ** 2 + 2 * #a ** 2 + #r ** 2 - 8 * #a - 3 * #r + 2); 0057 #a0 := 12 / (#a ** 2 * #r ** 2 + 3 * #a ** 2 * #r + 2 * #a * #r ** 2 + 2 * #a ** 2 + #r ** 2 - 8 * #a - 3 * #r + 2); 0058 #a1 := (12 * #a + 6 * #r + 6 * #a * #r - 12) / (#a ** 2 * #r ** 2 + 3 * #a ** 2 * #r + 2 * #a * #r ** 2 + 2 * #a ** 2 + #r ** 2 - 8 * #a - 3 * #r + 2); 0059 #a2 := 1; 0060 //sys2:=tf(k*[b0 b1 b2],[a0 a1 a2],T,'Variable', 'z**(-1)'); 0061 //sysdfod:=sys2; 0062 //przypisanie wyniku: 0063 #CFE_licz[0] := #b0; 0064 #CFE_licz[1] := #b1; 0065 #CFE_licz[2] := #b2; 0066 // 0067 #CFE_mian[0] := #a0; 0068 #CFE_mian[1] := #a1; 0069 #CFE_mian[2] := #a2; 0070 ELSIF (#n=3) THEN 0071 #b0 := 120 / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0072 #b1 := - (60 * #r + 180 - 180 * #a + 60 * #a * #r) / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0073 #b2 := - (-72 - 72 * #a ** 2 - 24 * #a * #r ** 2 + 216 * #a - 60 * #r - 12 * #r ** 2 - 12 * #a ** 2 * #r ** 2 + 60 * #a ** 2 * #r) / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0074 #b3 := - (11 * #r + 6 * #r ** 2 + #r ** 3 + 6 * #a * #r ** 2 - 6 * #a ** 2 * #r ** 2 - 27 * #a * #r - 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + #a ** 3 * #r ** 3 + 3 * #a * #r ** 3 + 11 * #a ** 3 * #r - 27 * #a ** 2 * #r - 6 * #a ** 3 - 54 * #a + 6 + 54 * #a ** 2) / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0075 #a0 := 120 / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0076 #a1 := (60 * #r - 180 + 180 * #a + 60 * #a * #r) / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0077 #a2 := (72 + 72 * #a ** 2 + 24 * #a * #r ** 2 - 216 * #a - 60 * #r + 12 * #r ** 2 + 12 * #a ** 2 * #r ** 2 + 60 * #a ** 2 * #r) / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0078 #a3 := 1; 0079 //przypisanie wyniku: 0080 #CFE_licz[0] := #b0; 0081 #CFE_licz[1] := #b1; 0082 #CFE_licz[2] := #b2; 0083 #CFE_licz[3] := #b3; 0084 // 0085 #CFE_mian[0] := #a0; 0086 #CFE_mian[1] := #a1; 0087 #CFE_mian[2] := #a2; 0088 #CFE_mian[3] := #a3; 0089 //sys3:=tf(k*[b0 b1 b2 b3],[a0 a1 a2 a3],T,'Variable', 'z**(-1)'); 0090 //sysdfod:=sys3; 0091 ELSIF (#n=4) THEN 0092 #b0 := 1680 / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); 0093 #b1 := (3360 * #a - 840 * #r - 840 * #a * #r - 3360) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - </pre>		

Totally Integrated Automation Portal		
	<pre> 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); </pre>	
0094	<pre> #b2 := (-5760 * #a + 2160 * #a ** 2 + 2160 + 360 * #a * #r ** 2 + 1260 * #r + 180 * #r ** 2 - 1260 * #a ** 2 * #r + 180 * #a ** 2 * #r ** 2) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); </pre>	
0095	<pre> #b3 := (-520 * #r + 960 * #a ** 2 * #r - 480 - 60 * #a ** 2 * #r ** 3 + 480 * #a ** 3 - 60 * #a * #r ** 3 + 180 * #a ** 3 * #r ** 2 - 520 * #a ** 3 * #r + 180 * #a ** 2 * #r ** 2 + 2880 * #a - 180 * #r ** 2 - 180 * #a * #r ** 2 - 20 * #r ** 3 + 960 * #a * #r - 20 * #a ** 3 * #r ** 3 - 2880 * #a ** 2) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); </pre>	
0096	<pre> #b4 := (10 * #r ** 3 - 40 * #a * #r ** 2 + #r ** 4 - 150 * #a ** 2 * #r ** 2 + 35 * #r ** 2 - 384 * #a ** 3 - 384 * #a + 50 * #r - 320 * #a * #r + 24 + 20 * #a * #r ** 3 + #a ** 4 * #r ** 4 - 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 - 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 - 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 864 * #a ** 2 + 24 * #a ** 4 + 320 * #a ** 3 * #r) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); </pre>	
0097	<pre> #a0 := 1680 / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); </pre>	
0098	<pre> #a1 := (3360 * #a + 840 * #r + 840 * #a * #r - 3360) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); </pre>	
0099	<pre> #a2 := (-5760 * #a + 2160 * #a ** 2 + 2160 + 360 * #a * #r ** 2 - 1260 * #r + 180 * #r ** 2 + 1260 * #a ** 2 * #r + 180 * #a ** 2 * #r ** 2) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); </pre>	
0100	<pre> #a3 := (520 * #r - 960 * #a ** 2 * #r - 480 + 60 * #a ** 2 * #r ** 3 + 480 * #a ** 3 + 60 * #a * #r ** 3 + 180 * #a ** 3 * #r ** 2 + 520 * #a ** 3 * #r + 180 * #a ** 2 * #r ** 2 + 2880 * #a - 180 * #r ** 2 - 180 * #a * #r ** 2 + 20 * #r ** 3 - 960 * #a * #r + 20 * #a ** 3 * #r ** 3 - 2880 * #a ** 2) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); </pre>	
0101	<pre> #a4 := 1; //przypisanie wyniku: #CFE_licz[0] := #b0; #CFE_licz[1] := #b1; #CFE_licz[2] := #b2; #CFE_licz[3] := #b3; #CFE_licz[4] := #b4; // #CFE_mian[0] := #a0; #CFE_mian[1] := #a1; #CFE_mian[2] := #a2; #CFE_mian[3] := #a3; #CFE_mian[4] := #a4; //sys4:=tf(k*[b0 b1 b2 b3 b4],[a0 a1 a2 a3 a4],T,'Variable', 'z**-1'); //sysdfod:=sys4; </pre>	
0116	<pre> ELSIF (#n=5) THEN #b0 := 30240 / (#a ** 5 * #r ** 5 + 15 * #a ** 5 * #r ** 4 + 5 * #a ** 4 * #r ** 5 + 85 * #a ** 5 * #r ** 3 + 45 * #a ** 4 * #r ** 4 + 10 * #a ** 3 * #r ** 5 + 225 * #a ** 5 * #r ** 2 + 5 * #a ** 4 * #r ** 3 + 30 * #a ** 3 * #r ** 4 + 10 * #a ** 2 * #r ** 5 + 274 * #a ** 5 * #r - 1005 * #a ** 4 * #r ** 2 - 410 * #a ** 3 * #r ** 3 - 30 * #a ** 2 * #r ** 4 + 5 * #a * #r ** 5 + 120 * #a ** 5 - 3250 * #a ** 4 * #r - 1230 * #a ** 3 * #r ** 2 - 410 * #a ** 2 * #r ** 3 - 45 * #a * #r ** 4 + #r ** 5 - 3000 * #a ** 4 + 4000 * #a ** 3 * #r + 1230 * #a ** 2 * #r ** 2 + 5 * #a * #r ** 3 - 15 * #r ** 4 + 12000 * #a ** 3 + 4000 * #a ** 2 * #r + 1005 * #a * #r ** 2 + 85 * #r ** 3 - 12000 * #a ** 2 - 3250 * #a * #r - 225 * #r ** 2 + 3000 * #a + 274 * #r - 120); </pre>	
0118	<pre> #b1 := - (-75600 * #a + 15120 * #r + 15120 * #a * #r + 75600) / (#a ** 5 * #r ** 5 + 15 * #a ** 5 * #r ** 4 + 5 * #a ** 4 * #r ** 5 + 85 * #a ** 5 * #r ** 3 + 45 * #a ** 4 * #r ** 4 + 10 * #a ** 3 * #r ** 5 + 225 * #a ** 5 * #r ** 2 + 5 * #a ** 4 * #r ** 3 + 30 * #a ** 3 * #r ** 4 + 10 * #a ** 2 * #r ** 5 + 274 * #a ** 5 * #r - 1005 * #a ** 4 * #r ** 2 - 410 * #a ** 3 * #r ** 3 - 30 * #a ** 2 * #r ** 4 + 5 * #a * #r ** 5 + 120 * #a ** 5 - 3250 * #a ** 4 * #r - 1230 * #a ** 3 * #r ** 2 - 410 * #a ** 2 * #r ** 3 - 45 * #a * #r ** 4 + #r ** 5 - 3000 * #a ** 4 + 4000 * #a ** 3 * #r + 1230 * #a ** 2 * #r ** 2 + 5 * #a * #r ** 3 - 15 * #r ** 4 + 12000 * #a ** 3 + 4000 * #a ** 2 * #r + 1005 * #a * #r ** 2 + 85 * #r ** 3 - 12000 * #a ** 2 - 3250 * #a * #r - 225 * #r ** 2 + 3000 * #a + 274 * #r - 120); </pre>	
0119	<pre> #b2 := - (168000 * #a - 30240 * #r - 67200 - 3360 * #a ** 2 * #r ** 2 - 3360 * #r ** 2 - 67200 * #a ** 2 - 6720 * #a * #r ** 2 + 30240 * #a ** 2 * #r) / (#a ** 5 * #r ** 5 + 15 * #a ** 5 * #r ** 4 + 5 * #a ** 4 * #r ** 5 + 85 * #a ** 5 * #r ** 3 + 45 * #a ** 4 * #r ** 4 + 10 * #a ** 3 * #r ** 5 + 225 * #a ** 5 * #r ** 2 + 5 * #a ** 4 * #r ** 3 + 30 * #a ** 3 * #r ** 4 + 10 * #a ** 2 * #r ** 5 + 274 * #a ** 5 * #r - 1005 * #a ** 4 * #r ** 2 - 410 * #a ** 3 * #r ** 3 - 30 * #a ** 2 * #r ** 4 + 5 * #a * #r ** 5 + 120 * #a ** 5 - 3250 * #a ** 4 * #r - 1230 * #a ** 3 * #r ** 2 - 410 * #a ** 2 * #r ** 3 - 45 * #a * #r ** 4 + #r ** 5 - 3000 * #a ** 4 + 4000 * #a ** 3 * #r + 1230 * #a ** 2 * #r ** 2 + 5 * #a * #r ** 3 - 15 * #r ** 4 + 12000 * #a ** 3 + 4000 * #a ** 2 * #r + 1005 * #a * #r ** 2 + 85 * #r ** 3 - 12000 * #a ** 2 - 3250 * #a * #r - 225 * #r ** 2 + 3000 * #a + 274 * #r - 120); </pre>	
0120	<pre> #b3 := - (5040 * #a * #r ** 2 - 31500 * #a ** 2 * #r - 5040 * #a ** 2 * #r ** 2 - 25200 * #a ** 3 + 19740 * #a ** 3 * #r + 5040 * #r ** 2 + 1260 * #a * #r ** 3 + 126000 * #a ** 2 - 31500 * #a * #r + 420 * #a ** 3 * #r ** 3 + 19740 * #r + 420 * #r ** 3 + 1260 * #a ** 2 * #r ** 3 + 25200 - 5040 * #a ** 3 * #r ** 2 - 126000 * #a) / (#a ** 5 * #r ** 5 + 15 * #a ** 5 * #r ** 4 + 5 * #a ** 4 * #r ** 5 + 85 * #a ** 5 * #r ** 3 + 45 * #a ** 4 * #r ** 4 + 10 * #a ** 3 * #r ** 5 + 225 * #a ** 5 * #r ** 2 + 5 * #a ** 4 * #r ** 3 + 30 * #a ** 3 * #r </pre>	


```

0142 #CFE_mian[5] := #a5;
0143 //sys5:=tf(k*[b0 b1 b2 b3 b4 b5],[a0 a1 a2 a3 a4 a5],T,'Variable', 'z**'-1');
0144 //sysdfod:=sys5;
0145 END_IF;
0146 //
0147
0148
0149
0150

```

Symbol	Address	Type	Comment
#a		Real	współczynnik
#a0		Real	
#a1		Real	
#a2		Real	
#a3		Real	
#a4		Real	
#a5		Real	
#b0		Real	
#b1		Real	
#b2		Real	
#b3		Real	
#b4		Real	
#b5		Real	
#CFE_licz		Array	
#CFE_mian		Array	
#k		Real	współczynnik
#m		Int	wskaznik petli do inicjalizacji
#n		USInt	rzad aproksymacji
#r		Real	rzad ułamkowy do zamodelowania
#T		Real	okres probkowania do obliczen

test_FO_s1200 / PLC_1 [CPU 1212C AC/DC/Rly] / Program blocks

CFE_DB [DB2]

CFE_DB Properties

General

Name	CFE_DB	Number	2	Type	DB	Language	DB
-------------	--------	---------------	---	-------------	----	-----------------	----

Numbering automatic

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

CFE_DB

Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Input							
Uin	Real	1.0	False	True	True	False	sygnał wejściowy
▼ wm	Array[0..5] of Real		True	True	True	False	współczynniki licznika CFE (?)
wm[0]	Real	0.0	True	True	True	False	
wm[1]	Real	0.0	True	True	True	False	
wm[2]	Real	0.0	True	True	True	False	
wm[3]	Real	0.0	True	True	True	False	
wm[4]	Real	0.0	True	True	True	False	
wm[5]	Real	0.0	True	True	True	False	
▼ vm	Array[0..5] of Real		True	True	True	False	współczynniki mianownika CFE (?)
vm[0]	Real	0.0	True	True	True	False	
vm[1]	Real	0.0	True	True	True	False	
vm[2]	Real	0.0	True	True	True	False	
vm[3]	Real	0.0	True	True	True	False	
vm[4]	Real	0.0	True	True	True	False	
vm[5]	Real	0.0	True	True	True	False	
reset	Bool	false	False	True	True	False	
▼ Output							
licznik	Int	0	False	True	True	False	
Yout	Real	0.0	False	True	True	False	sygnał wyjściowy
InOut							
▼ Static							
▼ y	Array[0..5] of Real		False	True	True	False	
y[0]	Real	0.0	False	True	True	False	
y[1]	Real	0.0	False	True	True	False	
y[2]	Real	0.0	False	True	True	False	
y[3]	Real	0.0	False	True	True	False	
y[4]	Real	0.0	False	True	True	False	
y[5]	Real	0.0	False	True	True	False	
test	USInt	0	False	True	True	False	
▼ u	Array[0..5] of Real		False	True	True	False	
u[0]	Real	0.0	False	True	True	False	
u[1]	Real	0.0	False	True	True	False	
u[2]	Real	0.0	False	True	True	False	
u[3]	Real	0.0	False	True	True	False	
u[4]	Real	0.0	False	True	True	False	
u[5]	Real	0.0	False	True	True	False	

test_FO_s1200 / PLC_1 [CPU 1212C AC/DC/Rly] / Program blocks

CFE [FB1]

CFE Properties

General

Name	CFE	Number	1	Type	FB	Language	SCL
------	-----	--------	---	------	----	----------	-----

Numbering automatic

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

CFE

Name	Data type	Default value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Input							
Uin	Real	1.0	Non-retain	True	True	False	sygnał wejściowy
▼ wm	Array[0..5] of Real		Retain	True	True	False	współczynniki licznika CFE (?)
wm[0]	Real	0.0	Retain	True	True	False	
wm[1]	Real	0.0	Retain	True	True	False	
wm[2]	Real	0.0	Retain	True	True	False	
wm[3]	Real	0.0	Retain	True	True	False	
wm[4]	Real	0.0	Retain	True	True	False	
wm[5]	Real	0.0	Retain	True	True	False	
▼ vm	Array[0..5] of Real		Retain	True	True	False	współczynniki mianownika CFE (?)
vm[0]	Real	0.0	Retain	True	True	False	
vm[1]	Real	0.0	Retain	True	True	False	
vm[2]	Real	0.0	Retain	True	True	False	
vm[3]	Real	0.0	Retain	True	True	False	
vm[4]	Real	0.0	Retain	True	True	False	
vm[5]	Real	0.0	Retain	True	True	False	
reset	Bool	false	Non-retain	True	True	False	
▼ Output							
licznik	Int	0	Non-retain	True	True	False	
Yout	Real	0.0	Non-retain	True	True	False	sygnał wyjściowy
InOut							
▼ Static							
▼ y	Array[0..5] of Real		Non-retain	True	True	False	
y[0]	Real	0.0	Non-retain	True	True	False	
y[1]	Real	0.0	Non-retain	True	True	False	
y[2]	Real	0.0	Non-retain	True	True	False	
y[3]	Real	0.0	Non-retain	True	True	False	
y[4]	Real	0.0	Non-retain	True	True	False	
y[5]	Real	0.0	Non-retain	True	True	False	
test	USInt	0	Non-retain	True	True	False	
▼ u	Array[0..5] of Real		Non-retain	True	True	False	
u[0]	Real	0.0	Non-retain	True	True	False	
u[1]	Real	0.0	Non-retain	True	True	False	
u[2]	Real	0.0	Non-retain	True	True	False	
u[3]	Real	0.0	Non-retain	True	True	False	
u[4]	Real	0.0	Non-retain	True	True	False	
u[5]	Real	0.0	Non-retain	True	True	False	
▼ Temp							
wsp_0	Real						
n	Int						
Constant							

```

0001 //reset wektorów przy zmianie współczynników modelu (aktywowany zmienną M0.0 "oblicz parametry")
0002 IF #reset THEN
0003   FOR #n:=0 TO 5 DO
0004     #y[#n] := 0;
0005     #u[#n] := 0;
0006   END_FOR;
0007   #licznik := 0;
0008 END_IF;
0009 //inicjalizacja sterowania:
0010 #u[5] := #Uin;
0011 //Aktualizacja licznika dla podanego skoku:
0012 IF #Uin > 0 THEN
0013   #licznik := #licznik + 1;
0014 END_IF;
0015 //obliczenia:
0016 //
0017 #wsp_0 := 1 / #vm[0];
0018 #y[5] := #wsp_0 * (-#vm[1] * #y[4] - #vm[2] * #y[3] - #vm[3] * #y[2] - #vm[4] * #y[1] - #vm[5] * #y[0]
+ #wm[0] * #u[5] + #wm[1] * #u[4] + #wm[2] * #u[3] + #wm[3] * #u[2] + #wm[4] * #u[1] + #wm[5] * #u[0]);
0019 //wynik:
0020 #Yout := 10.0 * #y[5];
0021 //przesunięcie wektorów wyjść:
0022 #y[0] := #y[1];

```



```

0023 #y[1] := #y[2];
0024 #y[2] := #y[3];
0025 #y[3] := #y[4];
0026 #y[4] := #y[5];
0027 //przesunięcie wektorów sterowań:
0028 #u[0] := #u[1];
0029 #u[1] := #u[2];
0030 #u[2] := #u[3];
0031 #u[3] := #u[4];
0032 #u[4] := #u[5];
0033 //test działania bloku:
0034 IF #test < 10 THEN
0035     #test := #test + 1;
0036 ELSE
0037     #test := 1;
0038 END_IF;
0039
0040 //
    
```

Symbol	Address	Type	Comment
#licznik		Int	
#n		Int	
#reset		Bool	
#test		USInt	
#u		Array	
#Uin		Real	sygnał wejściowy
#vm		Array	współczynniki mianownika CFE (?)
#wm		Array	współczynniki licznika CFE (?)
#wsp_0		Real	
#y		Array	
#Yout		Real	sygnał wyjściowy

test_FO_s1200 / PLC_1 [CPU 1212C AC/DC/Rly] / Program blocks

CFE_wsp [DB1]

CFE_wsp Properties

General

Name	CFE_wsp	Number	1	Type	DB	Language	DB
------	---------	--------	---	------	----	----------	----

Numbering	automatic
-----------	-----------

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

CFE_wsp

Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Static							
alfa	Real	0.5	True	True	True	False	
a	Real	1.0	True	True	True	False	
M	USInt	5	True	True	True	False	
▼ CFE_l	Array[0..5] of Real		True	True	True	False	
CFE_l[0]	Real	32.0	True	True	True	False	
CFE_l[1]	Real	-16.0	True	True	True	False	
CFE_l[2]	Real	-32.0	True	True	True	False	
CFE_l[3]	Real	12.0	True	True	True	False	
CFE_l[4]	Real	6.0	True	True	True	False	
CFE_l[5]	Real	-1.0	True	True	True	False	
▼ CFE_m	Array[0..5] of Real		True	True	True	False	
CFE_m[0]	Real	32.0	True	True	True	False	
CFE_m[1]	Real	16.0	True	True	True	False	
CFE_m[2]	Real	-32.0	True	True	True	False	
CFE_m[3]	Real	-12.0	True	True	True	False	
CFE_m[4]	Real	6.0	True	True	True	False	
CFE_m[5]	Real	1.0	True	True	True	False	

test_FO_s1200 / PLC_1 [CPU 1212C AC/DC/Rly] / Program blocks

Main [OB1]

Main Properties

General

Name	Main	Number	1	Type	OB	Language	LAD
-------------	------	---------------	---	-------------	----	-----------------	-----

Numbering automatic

Information

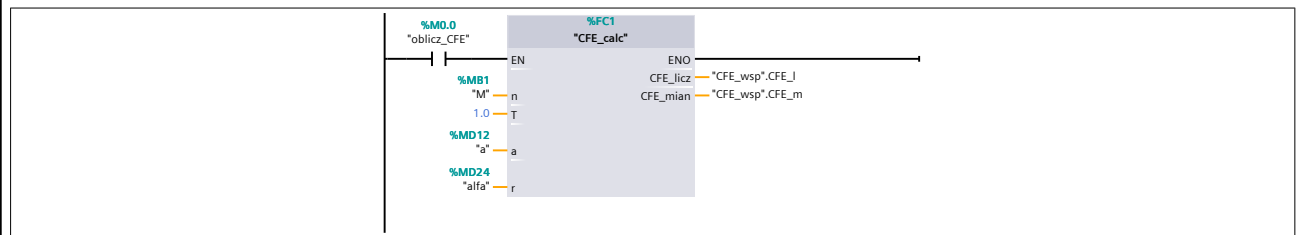
Title	program glowny	Author		Comment		Family	
Version	0.1	User-defined ID					

Main

Name	Data type	Default value	Comment
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
Temp			
Constant			

Network 1:

Obliczanie współczynników aproksymacji CFE: warunkowe, jeden raz po zmianie parametrów. Współczynniki są zapamiętywane a potem pobierane gotowe podczas obliczeń.



Symbol	Address	Type	Comment
"a"	%MD12	Real	
"alfa"	%MD24	Real	
"CFE_wsp".CFE_l		Array	
"CFE_wsp".CFE_m		Array	
"M"	%MB1	USInt	
"oblicz_CFE"	%M0.0	Bool	

test_FO_s1200 / PLC_1 [CPU 1212C AC/DC/Rly] / Program blocks

Cyclic interrupt [OB30]

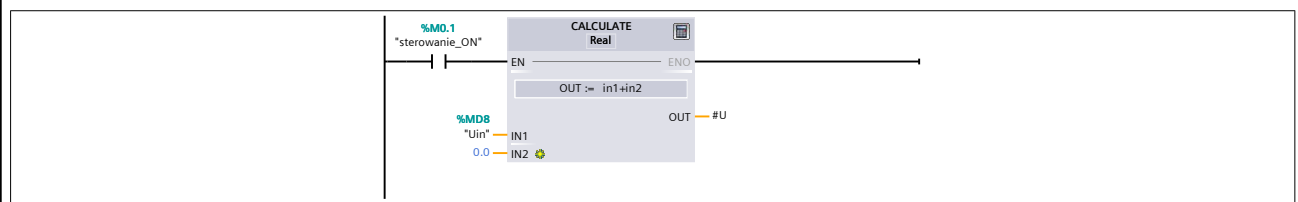
Cyclic interrupt Properties

General							
Name	Cyclic interrupt	Number	30	Type	OB	Language	LAD
Numbering	automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Cyclic interrupt

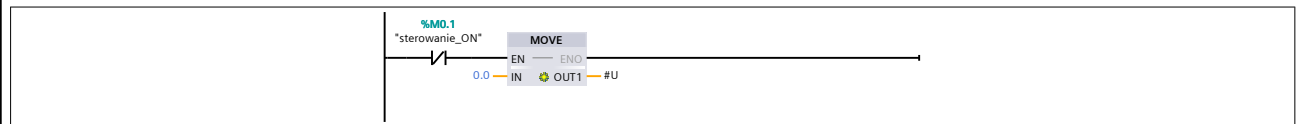
Name	Data type	Default value	Comment
▼ Input			
Initial_Call	Bool		Initial call of this OB
Event_Count	Int		Events discarded
▼ Temp			
U	Real		
Constant			

Network 1:



Symbol	Address	Type	Comment
"sterowanie_ON"	%M0.1	Bool	
"Uin"	%MD8	Real	
#U		Real	

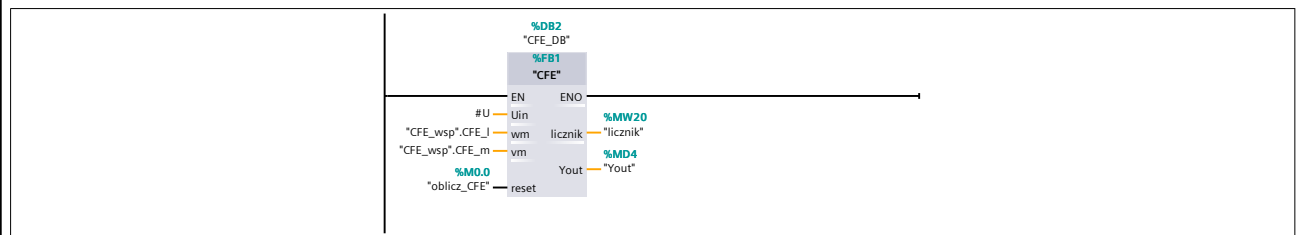
Network 2:



Symbol	Address	Type	Comment
"sterowanie_ON"	%M0.1	Bool	
#U		Real	

Network 3:

Wyznaczanie pochodnej lub całki ułamkowej z sygnału Uin. Parametry aproksymacji są pobierane z bloku danych CFE_wsp. Sygnał sterujący to Uin. Wyjście należy zebrać w aplikacji SCADA.



Symbol	Address	Type	Comment
"CFE_wsp".CFE_l		Array	
"CFE_wsp".CFE_m		Array	
"licznik"	%MW20	Int	
"oblicz_CFE"	%M0.0	Bool	
"Yout"	%MD4	Real	
#U		Real	

test_FO_s1200 / CFE [CPU 1511-1 PN] / Program blocks

CFE_calc [FC1]

CFE_calc Properties

General							
Name	CFE_calc	Number	1	Type	FC	Language	SCL
Numbering	automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

CFE_calc

Name	Data type	Default value	Comment
▼ Input			
n	USInt		rzad aproksymacji
T	Real		okres probkowania do obliczen
a	Real		współczynnik
r	Real		rzad ulamkowy do zamodelowania
▼ Output			
▼ CFE_licz	Array[0..5] of Real		
CFE_licz[0]	Real		
CFE_licz[1]	Real		
CFE_licz[2]	Real		
CFE_licz[3]	Real		
CFE_licz[4]	Real		
CFE_licz[5]	Real		
▼ CFE_mian	Array[0..5] of Real		
CFE_mian[0]	Real		
CFE_mian[1]	Real		
CFE_mian[2]	Real		
CFE_mian[3]	Real		
CFE_mian[4]	Real		
CFE_mian[5]	Real		
InOut			
▼ Temp			
k	Real		współczynnik
b0	Real		
b1	Real		
b2	Real		
b3	Real		
b4	Real		
b5	Real		
a0	Real		
a1	Real		
a2	Real		
a3	Real		
a4	Real		
a5	Real		
m	Int		wskaznik petli do inicjalizacji
Constant			
▼ Return			
CFE_calc	Void		

```

0001
0002
0003 //Funkcja różniczkowania i całkowania ułamkowego ze strony
0004 //Petrasa:
0005 //
0006 //http://people.tuke.sk/igor.podlubny/USU/matlab/petras/dfod1.m
0007 //function sysdfod=dfod1(n,T,a,r)
0008 // sysdfod=dfod1(n,T,a,r): new digital fractional order differentiator
0009 // AND integrator
0010 //
0011 // Output: :=>
0012 // Discrete system in the form OF the rational approximation - IIR filter
0013 // obtained BY continued fraction expansion OF a new generating function.
0014 //
0015 // Inputs: <:=
0016 // n: order OF truncation n:=(1 2 3 4 5)
0017 //
0018 //T: sampling period in [sec]
0019 // a: weighting factor between Euler AND Tustin rules ( 0 <:= a <:= 1 )
0020 // 0 - Euler rule, 1 - Tustin rule, 1/7 - Al-Alaoui rule, etc.
0021 // r: approximated fractional order (s**r), r is generally real number
0022 //
0023 // Author: Dr. Ivo Petras (ivo.petras@mail.com)
0024 // URL: http://ivopetras.tripod.com/
0025 //
0026 // Note: This approach is based on the original Al-Alaoui's work.
0027 //
0028 // Copyright (c), 2003.
0029 //

```

Totally Integrated Automation Portal		
<pre> 0030 //inicjalizacja: 0031 FOR #m := 0 TO 5 DO 0032 #CFE_licz[#m] := 0.0; 0033 #CFE_mian[#m] := 0.0; 0034 END_FOR; 0035 0036 0037 0038 #k := ((1+#a)/#T)**#r; 0039 0040 IF (#n=1) THEN 0041 #b0 := 2 / (#a * #r + #a + #r - 1); 0042 #b1 := - (#r + #a * #r - #a + 1) / (#a * #r + #a + #r - 1); 0043 #a0 := 2 / (#a * #r + #a + #r - 1); 0044 #a1 := 1; 0045 //przypisanie wyniku: 0046 #CFE_licz[0] := #b0; 0047 #CFE_licz[1] := #b1; 0048 // 0049 #CFE_mian[0] := #a0; 0050 #CFE_mian[1] := #a1; 0051 //sys1:=tf(k*[b0 b1],[a0 a1],T,'Variable', 'z**(-1)'); 0052 //sysdfod:=sys1; 0053 ELSIF (#n=2) THEN 0054 #b0 := 12 / (#a ** 2 * #r ** 2 + 3 * #a ** 2 * #r + 2 * #a * #r ** 2 + 2 * #a ** 2 + #r ** 2 - 8 * #a - 3 * #r + 2); 0055 #b1 := (12 * #a - 6 * #r - 6 * #a * #r - 12) / (#a ** 2 * #r ** 2 + 3 * #a ** 2 * #r + 2 * #a * #r ** 2 + 2 * #a ** 2 + #r ** 2 - 8 * #a - 3 * #r + 2); 0056 #b2 := (#a ** 2 * #r ** 2 - 3 * #a ** 2 * #r + 2 * #a * #r ** 2 + #r ** 2 - 8 * #a + 3 * #r + 2 * #a ** 2 + 2) / (#a ** 2 * #r ** 2 + 3 * #a ** 2 * #r + 2 * #a * #r ** 2 + 2 * #a ** 2 + #r ** 2 - 8 * #a - 3 * #r + 2); 0057 #a0 := 12 / (#a ** 2 * #r ** 2 + 3 * #a ** 2 * #r + 2 * #a * #r ** 2 + 2 * #a ** 2 + #r ** 2 - 8 * #a - 3 * #r + 2); 0058 #a1 := (12 * #a + 6 * #r + 6 * #a * #r - 12) / (#a ** 2 * #r ** 2 + 3 * #a ** 2 * #r + 2 * #a * #r ** 2 + 2 * #a ** 2 + #r ** 2 - 8 * #a - 3 * #r + 2); 0059 #a2 := 1; 0060 //sys2:=tf(k*[b0 b1 b2],[a0 a1 a2],T,'Variable', 'z**(-1)'); 0061 //sysdfod:=sys2; 0062 //przypisanie wyniku: 0063 #CFE_licz[0] := #b0; 0064 #CFE_licz[1] := #b1; 0065 #CFE_licz[2] := #b2; 0066 // 0067 #CFE_mian[0] := #a0; 0068 #CFE_mian[1] := #a1; 0069 #CFE_mian[2] := #a2; 0070 ELSIF (#n=3) THEN 0071 #b0 := 120 / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0072 #b1 := - (60 * #r + 180 - 180 * #a + 60 * #a * #r) / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0073 #b2 := - (-72 - 72 * #a ** 2 - 24 * #a * #r ** 2 + 216 * #a - 60 * #r - 12 * #r ** 2 - 12 * #a ** 2 * #r ** 2 + 60 * #a ** 2 * #r) / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0074 #b3 := - (11 * #r + 6 * #r ** 2 + #r ** 3 + 6 * #a * #r ** 2 - 6 * #a ** 2 * #r ** 2 - 27 * #a * #r - 6 * #a ** 3 * #r ** 2 + 3 * #a * #r ** 3 + #a ** 3 * #r ** 3 + 3 * #a * #r ** 3 + 11 * #a ** 3 * #r - 27 * #a ** 2 * #r - 6 * #a ** 3 - 54 * #a + 6 + 54 * #a ** 2) / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0075 #a0 := 120 / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0076 #a1 := (60 * #r - 180 + 180 * #a + 60 * #a * #r) / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0077 #a2 := (72 + 72 * #a ** 2 + 24 * #a * #r ** 2 - 216 * #a - 60 * #r + 12 * #r ** 2 + 12 * #a ** 2 * #r ** 2 + 60 * #a ** 2 * #r) / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0078 #a3 := 1; 0079 //przypisanie wyniku: 0080 #CFE_licz[0] := #b0; 0081 #CFE_licz[1] := #b1; 0082 #CFE_licz[2] := #b2; 0083 #CFE_licz[3] := #b3; 0084 // 0085 #CFE_mian[0] := #a0; 0086 #CFE_mian[1] := #a1; 0087 #CFE_mian[2] := #a2; 0088 #CFE_mian[3] := #a3; 0089 //sys3:=tf(k*[b0 b1 b2 b3],[a0 a1 a2 a3],T,'Variable', 'z**(-1)'); 0090 //sysdfod:=sys3; 0091 ELSIF (#n=4) THEN 0092 #b0 := 1680 / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); 0093 #b1 := (3360 * #a - 840 * #r - 840 * #a * #r - 3360) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - </pre>		

Totally Integrated Automation Portal		
	<pre> 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); </pre>	
0094	<pre> #b2 := (-5760 * #a + 2160 * #a ** 2 + 2160 + 360 * #a * #r ** 2 + 1260 * #r + 180 * #r ** 2 - 1260 * #a ** 2 * #r + 180 * #a ** 2 * #r ** 2) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); </pre>	
0095	<pre> #b3 := (-520 * #r + 960 * #a ** 2 * #r - 480 - 60 * #a ** 2 * #r ** 3 + 480 * #a ** 3 - 60 * #a * #r ** 3 + 180 * #a ** 3 * #r ** 2 - 520 * #a ** 3 * #r + 180 * #a ** 2 * #r ** 2 + 2880 * #a - 180 * #r ** 2 - 180 * #a * #r ** 2 - 20 * #r ** 3 + 960 * #a * #r - 20 * #a ** 3 * #r ** 3 - 2880 * #a ** 2) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); </pre>	
0096	<pre> #b4 := (10 * #r ** 3 - 40 * #a * #r ** 2 + #r ** 4 - 150 * #a ** 2 * #r ** 2 + 35 * #r ** 2 - 384 * #a ** 3 - 384 * #a + 50 * #r - 320 * #a * #r + 24 + 20 * #a * #r ** 3 + #a ** 4 * #r ** 4 - 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 - 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 - 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 864 * #a ** 2 + 24 * #a ** 4 + 320 * #a ** 3 * #r) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); </pre>	
0097	<pre> #a0 := 1680 / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); </pre>	
0098	<pre> #a1 := (3360 * #a + 840 * #r + 840 * #a * #r - 3360) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); </pre>	
0099	<pre> #a2 := (-5760 * #a + 2160 * #a ** 2 + 2160 + 360 * #a * #r ** 2 - 1260 * #r + 180 * #r ** 2 + 1260 * #a ** 2 * #r + 180 * #a ** 2 * #r ** 2) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); </pre>	
0100	<pre> #a3 := (520 * #r - 960 * #a ** 2 * #r - 480 + 60 * #a ** 2 * #r ** 3 + 480 * #a ** 3 + 60 * #a * #r ** 3 + 180 * #a ** 3 * #r ** 2 + 520 * #a ** 3 * #r + 180 * #a ** 2 * #r ** 2 + 2880 * #a - 180 * #r ** 2 - 180 * #a * #r ** 2 + 20 * #r ** 3 - 960 * #a * #r + 20 * #a ** 3 * #r ** 3 - 2880 * #a ** 2) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); </pre>	
0101	<pre> #a4 := 1; //przypisanie wyniku: #CFE_licz[0] := #b0; #CFE_licz[1] := #b1; #CFE_licz[2] := #b2; #CFE_licz[3] := #b3; #CFE_licz[4] := #b4; // #CFE_mian[0] := #a0; #CFE_mian[1] := #a1; #CFE_mian[2] := #a2; #CFE_mian[3] := #a3; #CFE_mian[4] := #a4; //sys4:=tf(k*[b0 b1 b2 b3 b4],[a0 a1 a2 a3 a4],T,'Variable', 'z**-1'); //sysdfod:=sys4; </pre>	
0116	<pre> ELSIF (#n=5) THEN #b0 := 30240 / (#a ** 5 * #r ** 5 + 15 * #a ** 5 * #r ** 4 + 5 * #a ** 4 * #r ** 5 + 85 * #a ** 5 * #r ** 3 + 45 * #a ** 4 * #r ** 4 + 10 * #a ** 3 * #r ** 5 + 225 * #a ** 5 * #r ** 2 + 5 * #a ** 4 * #r ** 3 + 30 * #a ** 3 * #r ** 4 + 10 * #a ** 2 * #r ** 5 + 274 * #a ** 5 * #r - 1005 * #a ** 4 * #r ** 2 - 410 * #a ** 3 * #r ** 3 - 30 * #a ** 2 * #r ** 4 + 5 * #a * #r ** 5 + 120 * #a ** 5 - 3250 * #a ** 4 * #r - 1230 * #a ** 3 * #r ** 2 - 410 * #a ** 2 * #r ** 3 - 45 * #a * #r ** 4 + #r ** 5 - 3000 * #a ** 4 + 4000 * #a ** 3 * #r + 1230 * #a ** 2 * #r ** 2 + 5 * #a * #r ** 3 - 15 * #r ** 4 + 12000 * #a ** 3 + 4000 * #a ** 2 * #r + 1005 * #a * #r ** 2 + 85 * #r ** 3 - 12000 * #a ** 2 - 3250 * #a * #r - 225 * #r ** 2 + 3000 * #a + 274 * #r - 120); </pre>	
0118	<pre> #b1 := - (-75600 * #a + 15120 * #r + 15120 * #a * #r + 75600) / (#a ** 5 * #r ** 5 + 15 * #a ** 5 * #r ** 4 + 5 * #a ** 4 * #r ** 5 + 85 * #a ** 5 * #r ** 3 + 45 * #a ** 4 * #r ** 4 + 10 * #a ** 3 * #r ** 5 + 225 * #a ** 5 * #r ** 2 + 5 * #a ** 4 * #r ** 3 + 30 * #a ** 3 * #r ** 4 + 10 * #a ** 2 * #r ** 5 + 274 * #a ** 5 * #r - 1005 * #a ** 4 * #r ** 2 - 410 * #a ** 3 * #r ** 3 - 30 * #a ** 2 * #r ** 4 + 5 * #a * #r ** 5 + 120 * #a ** 5 - 3250 * #a ** 4 * #r - 1230 * #a ** 3 * #r ** 2 - 410 * #a ** 2 * #r ** 3 - 45 * #a * #r ** 4 + #r ** 5 - 3000 * #a ** 4 + 4000 * #a ** 3 * #r + 1230 * #a ** 2 * #r ** 2 + 5 * #a * #r ** 3 - 15 * #r ** 4 + 12000 * #a ** 3 + 4000 * #a ** 2 * #r + 1005 * #a * #r ** 2 + 85 * #r ** 3 - 12000 * #a ** 2 - 3250 * #a * #r - 225 * #r ** 2 + 3000 * #a + 274 * #r - 120); </pre>	
0119	<pre> #b2 := - (168000 * #a - 30240 * #r - 67200 - 3360 * #a ** 2 * #r ** 2 - 3360 * #r ** 2 - 67200 * #a ** 2 - 6720 * #a * #r ** 2 + 30240 * #a ** 2 * #r) / (#a ** 5 * #r ** 5 + 15 * #a ** 5 * #r ** 4 + 5 * #a ** 4 * #r ** 5 + 85 * #a ** 5 * #r ** 3 + 45 * #a ** 4 * #r ** 4 + 10 * #a ** 3 * #r ** 5 + 225 * #a ** 5 * #r ** 2 + 5 * #a ** 4 * #r ** 3 + 30 * #a ** 3 * #r ** 4 + 10 * #a ** 2 * #r ** 5 + 274 * #a ** 5 * #r - 1005 * #a ** 4 * #r ** 2 - 410 * #a ** 3 * #r ** 3 - 30 * #a ** 2 * #r ** 4 + 5 * #a * #r ** 5 + 120 * #a ** 5 - 3250 * #a ** 4 * #r - 1230 * #a ** 3 * #r ** 2 - 410 * #a ** 2 * #r ** 3 - 45 * #a * #r ** 4 + #r ** 5 - 3000 * #a ** 4 + 4000 * #a ** 3 * #r + 1230 * #a ** 2 * #r ** 2 + 5 * #a * #r ** 3 - 15 * #r ** 4 + 12000 * #a ** 3 + 4000 * #a ** 2 * #r + 1005 * #a * #r ** 2 + 85 * #r ** 3 - 12000 * #a ** 2 - 3250 * #a * #r - 225 * #r ** 2 + 3000 * #a + 274 * #r - 120); </pre>	
0120	<pre> #b3 := - (5040 * #a * #r ** 2 - 31500 * #a ** 2 * #r - 5040 * #a ** 2 * #r ** 2 - 25200 * #a ** 3 + 19740 * #a ** 3 * #r + 5040 * #r ** 2 + 1260 * #a * #r ** 3 + 126000 * #a ** 2 - 31500 * #a * #r + 420 * #a ** 3 * #r ** 3 + 19740 * #r + 420 * #r ** 3 + 1260 * #a ** 2 * #r ** 3 + 25200 - 5040 * #a ** 3 * #r ** 2 - 126000 * #a) / (#a ** 5 * #r ** 5 + 15 * #a ** 5 * #r ** 4 + 5 * #a ** 4 * #r ** 5 + 85 * #a ** 5 * #r ** 3 + 45 * #a ** 4 * #r ** 4 + 10 * #a ** 3 * #r ** 5 + 225 * #a ** 5 * #r ** 2 + 5 * #a ** 4 * #r ** 3 + 30 * #a ** 3 * #r </pre>	


```

0142 #CFE_mian[5] := #a5;
0143 //sys5:=tf(k*[b0 b1 b2 b3 b4 b5],[a0 a1 a2 a3 a4 a5],T,'Variable', 'z**'-1');
0144 //sysdfod:=sys5;
0145 END_IF;
0146 //
0147
0148
0149
0150

```

Symbol	Address	Type	Comment
#a		Real	współczynnik
#a0		Real	
#a1		Real	
#a2		Real	
#a3		Real	
#a4		Real	
#a5		Real	
#b0		Real	
#b1		Real	
#b2		Real	
#b3		Real	
#b4		Real	
#b5		Real	
#CFE_licz		Array	
#CFE_mian		Array	
#k		Real	współczynnik
#m		Int	wskaznik petli do inicjalizacji
#n		USInt	rzad aproksymacji
#r		Real	rzad ulamkowy do zamodelowania
#T		Real	okres probkowania do obliczen

test_FO_s1200 / CFE [CPU 1511-1 PN] / Program blocks

CFE_DB [DB2]

CFE_DB Properties

General

Name	CFE_DB	Number	2	Type	DB	Language	DB
-------------	--------	---------------	---	-------------	----	-----------------	----

Numbering automatic

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

CFE_DB

Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Input							
Uin	Real	1.0	False	True	True	False	sygnał wejściowy
▼ wm	Array[0..5] of Real		True	True	True	False	współczynniki licznika CFE (?)
wm[0]	Real	0.0	True	True	True	False	
wm[1]	Real	0.0	True	True	True	False	
wm[2]	Real	0.0	True	True	True	False	
wm[3]	Real	0.0	True	True	True	False	
wm[4]	Real	0.0	True	True	True	False	
wm[5]	Real	0.0	True	True	True	False	
▼ vm	Array[0..5] of Real		True	True	True	False	współczynniki mianownika CFE (?)
vm[0]	Real	0.0	True	True	True	False	
vm[1]	Real	0.0	True	True	True	False	
vm[2]	Real	0.0	True	True	True	False	
vm[3]	Real	0.0	True	True	True	False	
vm[4]	Real	0.0	True	True	True	False	
vm[5]	Real	0.0	True	True	True	False	
reset	Bool	false	False	True	True	False	
▼ Output							
licznik	Int	0	False	True	True	False	
Yout	Real	0.0	False	True	True	False	sygnał wyjściowy
InOut							
▼ Static							
▼ y	Array[0..5] of Real		False	True	True	False	
y[0]	Real	0.0	False	True	True	False	
y[1]	Real	0.0	False	True	True	False	
y[2]	Real	0.0	False	True	True	False	
y[3]	Real	0.0	False	True	True	False	
y[4]	Real	0.0	False	True	True	False	
y[5]	Real	0.0	False	True	True	False	
test	USInt	0	False	True	True	False	
▼ u	Array[0..5] of Real		False	True	True	False	
u[0]	Real	0.0	False	True	True	False	
u[1]	Real	0.0	False	True	True	False	
u[2]	Real	0.0	False	True	True	False	
u[3]	Real	0.0	False	True	True	False	
u[4]	Real	0.0	False	True	True	False	
u[5]	Real	0.0	False	True	True	False	

test_FO_s1200 / CFE [CPU 1511-1 PN] / Program blocks

CFE [FB1]

CFE Properties

General

Name	CFE	Number	1	Type	FB	Language	SCL
------	-----	--------	---	------	----	----------	-----

Numbering automatic

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

CFE

Name	Data type	Default value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Input							
Uin	Real	1.0	Non-retain	True	True	False	sygnał wejściowy
▼ wm	Array[0..5] of Real		Retain	True	True	False	współczynniki licznika CFE (?)
wm[0]	Real	0.0	Retain	True	True	False	
wm[1]	Real	0.0	Retain	True	True	False	
wm[2]	Real	0.0	Retain	True	True	False	
wm[3]	Real	0.0	Retain	True	True	False	
wm[4]	Real	0.0	Retain	True	True	False	
wm[5]	Real	0.0	Retain	True	True	False	
▼ vm	Array[0..5] of Real		Retain	True	True	False	współczynniki mianownika CFE (?)
vm[0]	Real	0.0	Retain	True	True	False	
vm[1]	Real	0.0	Retain	True	True	False	
vm[2]	Real	0.0	Retain	True	True	False	
vm[3]	Real	0.0	Retain	True	True	False	
vm[4]	Real	0.0	Retain	True	True	False	
vm[5]	Real	0.0	Retain	True	True	False	
reset	Bool	false	Non-retain	True	True	False	
▼ Output							
licznik	Int	0	Non-retain	True	True	False	
Yout	Real	0.0	Non-retain	True	True	False	sygnał wyjściowy
InOut							
▼ Static							
▼ y	Array[0..5] of Real		Non-retain	True	True	False	
y[0]	Real	0.0	Non-retain	True	True	False	
y[1]	Real	0.0	Non-retain	True	True	False	
y[2]	Real	0.0	Non-retain	True	True	False	
y[3]	Real	0.0	Non-retain	True	True	False	
y[4]	Real	0.0	Non-retain	True	True	False	
y[5]	Real	0.0	Non-retain	True	True	False	
test	USInt	0	Non-retain	True	True	False	
▼ u	Array[0..5] of Real		Non-retain	True	True	False	
u[0]	Real	0.0	Non-retain	True	True	False	
u[1]	Real	0.0	Non-retain	True	True	False	
u[2]	Real	0.0	Non-retain	True	True	False	
u[3]	Real	0.0	Non-retain	True	True	False	
u[4]	Real	0.0	Non-retain	True	True	False	
u[5]	Real	0.0	Non-retain	True	True	False	
▼ Temp							
wsp_0	Real						
n	Int						
Constant							

```

0001 //reset wektorów przy zmianie współczynników modelu (aktywowany zmienną M0.0 "oblicz parametry")
0002 IF #reset THEN
0003   FOR #n:=0 TO 5 DO
0004     #y[#n] := 0;
0005     #u[#n] := 0;
0006   END_FOR;
0007   #licznik := 0;
0008 END_IF;
0009 //inicjalizacja sterowania:
0010 #u[5] := #Uin;
0011 //Aktualizacja licznika dla podanego skoku:
0012 IF #Uin > 0 THEN
0013   #licznik := #licznik + 1;
0014 END_IF;
0015 //obliczenia:
0016 //
0017 #wsp_0 := 1 / #vm[0];
0018 #y[5] := #wsp_0 * (-#vm[1] * #y[4] - #vm[2] * #y[3] - #vm[3] * #y[2] - #vm[4] * #y[1] - #vm[5] * #y[0]
+ #wm[0] * #u[5] + #wm[1] * #u[4] + #wm[2] * #u[3] + #wm[3] * #u[2] + #wm[4] * #u[1] + #wm[5] * #u[0]);
0019 //wynik:
0020 #Yout := 10.0 * #y[5];
0021 //przesunięcie wektorów wyjść:
0022 #y[0] := #y[1];

```

```

0023 #y[1] := #y[2];
0024 #y[2] := #y[3];
0025 #y[3] := #y[4];
0026 #y[4] := #y[5];
0027 //przesunięcie wektorów sterowań:
0028 #u[0] := #u[1];
0029 #u[1] := #u[2];
0030 #u[2] := #u[3];
0031 #u[3] := #u[4];
0032 #u[4] := #u[5];
0033 //test działania bloku:
0034 IF #test < 10 THEN
0035     #test := #test + 1;
0036 ELSE
0037     #test := 1;
0038 END_IF;
0039
0040 //
    
```

Symbol	Address	Type	Comment
#licznik		Int	
#n		Int	
#reset		Bool	
#test		USInt	
#u		Array	
#Uin		Real	sygnał wejściowy
#vm		Array	współczynniki mianownika CFE (?)
#wm		Array	współczynniki licznika CFE (?)
#wsp_0		Real	
#y		Array	
#Yout		Real	sygnał wyjściowy

test_FO_s1200 / CFE [CPU 1511-1 PN] / Program blocks

CFE_wsp [DB1]

CFE_wsp Properties

General

Name	CFE_wsp	Number	1	Type	DB	Language	DB
-------------	---------	---------------	---	-------------	----	-----------------	----

Numbering	automatic
------------------	-----------

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

CFE_wsp

Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Static							
alfa	Real	0.5	True	True	True	False	
a	Real	1.0	True	True	True	False	
M	USInt	5	True	True	True	False	
▼ CFE_l	Array[0..5] of Real		True	True	True	False	
CFE_l[0]	Real	32.0	True	True	True	False	
CFE_l[1]	Real	-16.0	True	True	True	False	
CFE_l[2]	Real	-32.0	True	True	True	False	
CFE_l[3]	Real	12.0	True	True	True	False	
CFE_l[4]	Real	6.0	True	True	True	False	
CFE_l[5]	Real	-1.0	True	True	True	False	
▼ CFE_m	Array[0..5] of Real		True	True	True	False	
CFE_m[0]	Real	32.0	True	True	True	False	
CFE_m[1]	Real	16.0	True	True	True	False	
CFE_m[2]	Real	-32.0	True	True	True	False	
CFE_m[3]	Real	-12.0	True	True	True	False	
CFE_m[4]	Real	6.0	True	True	True	False	
CFE_m[5]	Real	1.0	True	True	True	False	

test_FO_s1200 / CFE [CPU 1511-1 PN] / Program blocks

Cyclic interrupt [OB30]

Cyclic interrupt Properties

General

Name	Cyclic interrupt	Number	30	Type	OB	Language	LAD
Numbering	automatic						

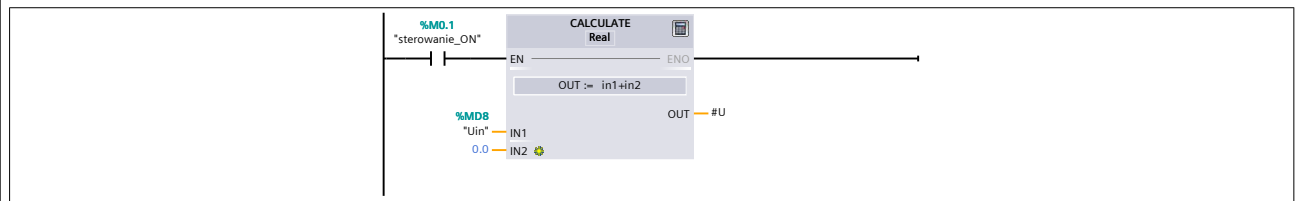
Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Cyclic interrupt

Name	Data type	Default value	Comment
▼ Input			
Initial_Call	Bool		Initial call of this OB
Event_Count	Int		Events discarded
▼ Temp			
U	Real		
Constant			

Network 1:



Symbol	Address	Type	Comment
"sterowanie_ON"	%M0.1	Bool	
"Uin"	%MD8	Real	
#U		Real	

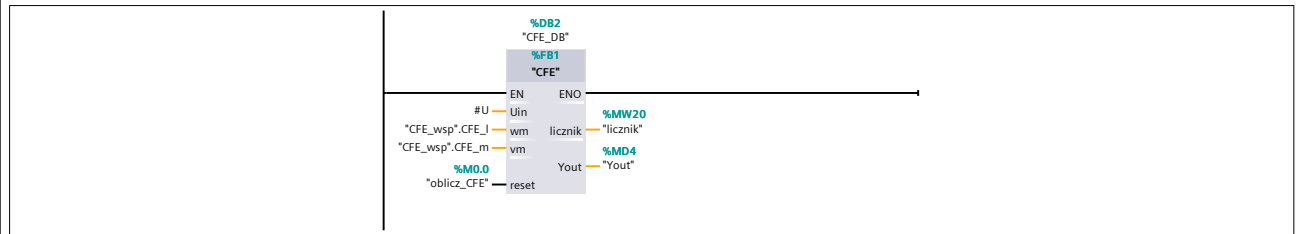
Network 2:



Symbol	Address	Type	Comment
"sterowanie_ON"	%M0.1	Bool	
#U		Real	

Network 3:

Wyznaczanie pochodnej lub całki ułamkowej z sygnału Uin. Parametry aproksymacji są pobierane z bloku danych CFE_wsp. Sygnał sterujący to Uin. Wyjście należy zebrać w aplikacji SCADA.



Symbol	Address	Type	Comment
"CFE_wsp".CFE_l		Array	
"CFE_wsp".CFE_m		Array	
"licznik"	%MW20	Int	
"oblicz_CFE"	%M0.0	Bool	
"Yout"	%MD4	Real	
#U		Real	

test_FO_s1200 / CFE [CPU 1511-1 PN] / Program blocks

Main [OB1]

Main Properties

General

Name	Main	Number	1	Type	OB	Language	LAD
-------------	------	---------------	---	-------------	----	-----------------	-----

Numbering automatic

Information

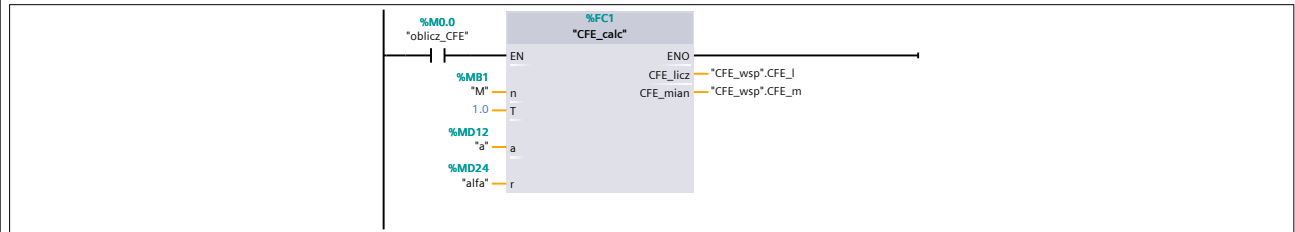
Title	program glowny	Author		Comment		Family	
Version	0.1	User-defined ID					

Main

Name	Data type	Default value	Comment
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
▼ Temp			
RT_INFO	Int		
czas_wykonania_temp	Real		
różnica_aktualny_średni_czas	Real		
czas_wykonania_temp_1	Real		
OB30_RT_INFO	Int		
OB30_czas_wykonania_temp	Real		
OB30_różnica_aktualny_średni_czas	Real		
OB30_czas_wykonania_temp_1	Real		
Constant			

Network 1:

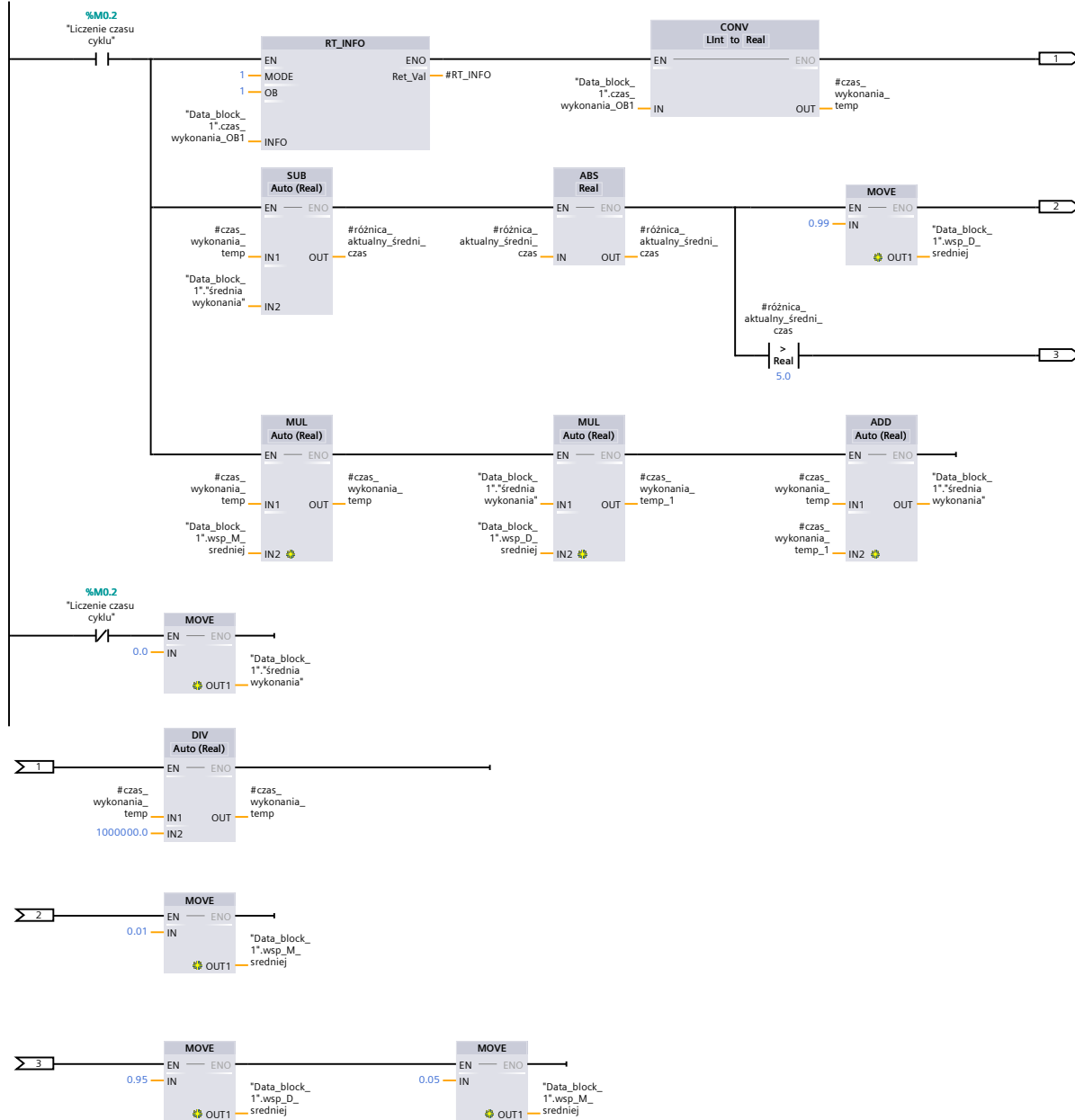
Obliczanie współczynników aproksymacji CFE: warunkowe, jeden raz po zmianie parametrów. Współczynniki są zapamiętywane a potem pobierane gotowe podczas obliczeń.



Symbol	Address	Type	Comment
"a"	%MD12	Real	
"alfa"	%MD24	Real	
"CFE_wsp".CFE_l		Array	
"CFE_wsp".CFE_m		Array	
"M"	%MB1	USInt	
"oblicz_CFE"	%M0.0	Bool	

Network 2: OB1

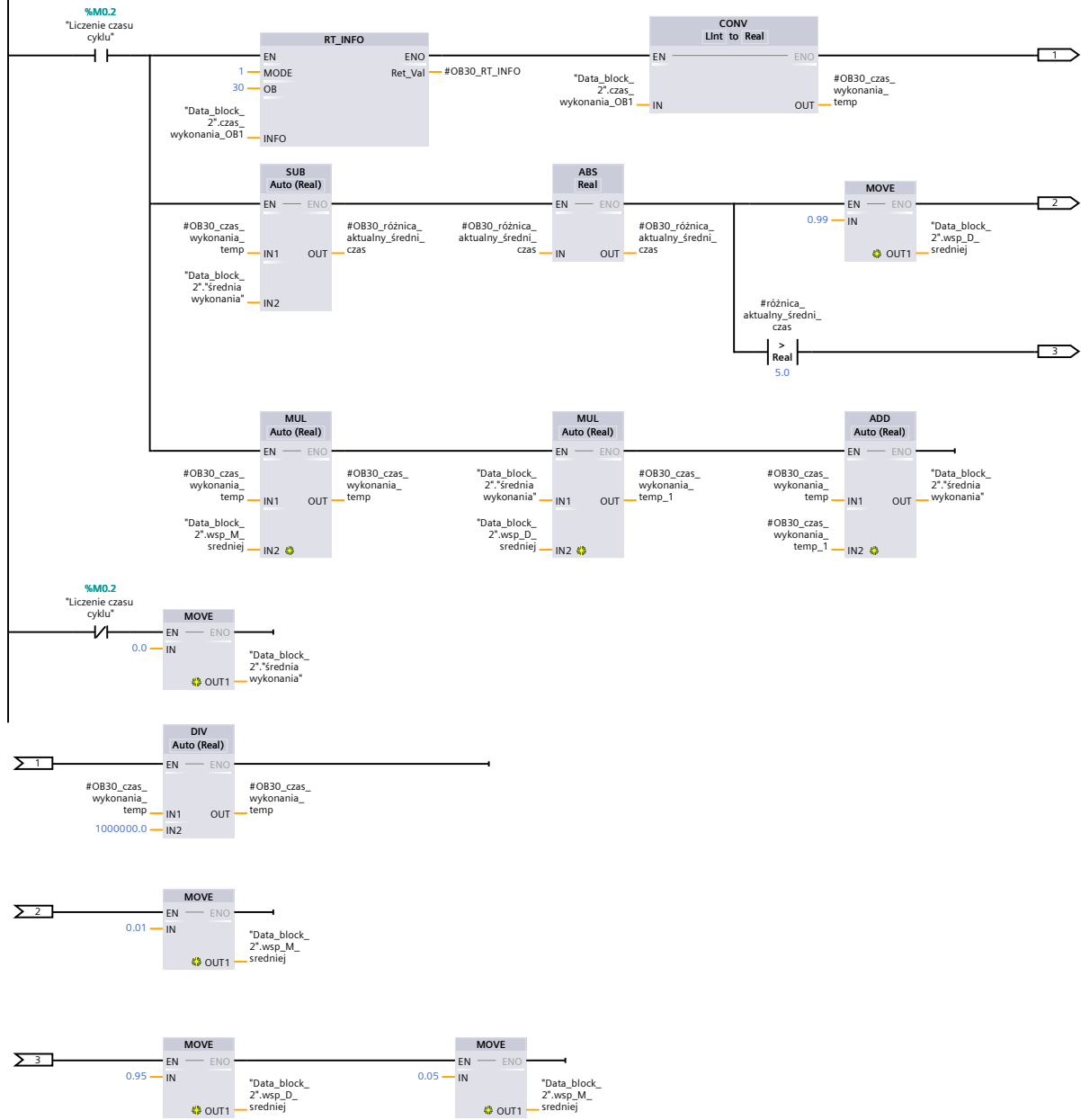
Network 2: OB1



Symbol	Address	Type	Comment
"Data_block_1".srednia_wykonania		Real	
"Data_block_1".czas_wykonania_OB1		LTime	
"Data_block_1".wsp_D_sredniej		Real	
"Data_block_1".wsp_M_sredniej		Real	
"Liczenie czasu cyklu"	%M0.2	Bool	
#czas_wykonania_temp		Real	
#czas_wykonania_temp_1		Real	
#różnica_aktualny_sredni_czas		Real	
#RT_INFO		Int	

Network 3: OB30

Network 3: OB30



Symbol	Address	Type	Comment
"Data_block_2".średnia wykonania		Real	
"Data_block_2".czas_wykonania_OB1		LTime	
"Data_block_2".wsp_D_średniej		Real	
"Data_block_2".wsp_M_średniej		Real	
"Liczenie czasu cyklu"	%M0.2	Bool	
#OB30_czas_wykonania_temp		Real	
#OB30_czas_wykonania_temp_1		Real	
#OB30_różnica_aktualny_średni_czas		Real	
#OB30_RT_INFO		Int	
#różnica_aktualny_średni_czas		Real	

test_FO_s1200 / PLC_1 [CPU 1212C AC/DC/Rly] / Program blocks

CFE_wsp [DB1]

CFE_wsp Properties

General

Name	CFE_wsp	Number	1	Type	DB	Language	DB
------	---------	--------	---	------	----	----------	----

Numbering	automatic
-----------	-----------

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

CFE_wsp

Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Static							
alfa	Real	0.5	True	True	True	False	
a	Real	1.0	True	True	True	False	
M	USInt	5	True	True	True	False	
▼ CFE_l	Array[0..5] of Real		True	True	True	False	
CFE_l[0]	Real	32.0	True	True	True	False	
CFE_l[1]	Real	-16.0	True	True	True	False	
CFE_l[2]	Real	-32.0	True	True	True	False	
CFE_l[3]	Real	12.0	True	True	True	False	
CFE_l[4]	Real	6.0	True	True	True	False	
CFE_l[5]	Real	-1.0	True	True	True	False	
▼ CFE_m	Array[0..5] of Real		True	True	True	False	
CFE_m[0]	Real	32.0	True	True	True	False	
CFE_m[1]	Real	16.0	True	True	True	False	
CFE_m[2]	Real	-32.0	True	True	True	False	
CFE_m[3]	Real	-12.0	True	True	True	False	
CFE_m[4]	Real	6.0	True	True	True	False	
CFE_m[5]	Real	1.0	True	True	True	False	

test_FO_s1200 / PLC_1 [CPU 1212C AC/DC/Rly] / Program blocks

CFE_DB [DB2]

CFE_DB Properties

General

Name	CFE_DB	Number	2	Type	DB	Language	DB
-------------	--------	---------------	---	-------------	----	-----------------	----

Numbering automatic

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

CFE_DB

Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Input							
Uin	Real	1.0	False	True	True	False	sygnał wejściowy
▼ wm	Array[0..5] of Real		True	True	True	False	współczynniki licznika CFE (?)
wm[0]	Real	0.0	True	True	True	False	
wm[1]	Real	0.0	True	True	True	False	
wm[2]	Real	0.0	True	True	True	False	
wm[3]	Real	0.0	True	True	True	False	
wm[4]	Real	0.0	True	True	True	False	
wm[5]	Real	0.0	True	True	True	False	
▼ vm	Array[0..5] of Real		True	True	True	False	współczynniki mianownika CFE (?)
vm[0]	Real	0.0	True	True	True	False	
vm[1]	Real	0.0	True	True	True	False	
vm[2]	Real	0.0	True	True	True	False	
vm[3]	Real	0.0	True	True	True	False	
vm[4]	Real	0.0	True	True	True	False	
vm[5]	Real	0.0	True	True	True	False	
reset	Bool	false	False	True	True	False	
▼ Output							
licznik	Int	0	False	True	True	False	
Yout	Real	0.0	False	True	True	False	sygnał wyjściowy
InOut							
▼ Static							
▼ y	Array[0..5] of Real		False	True	True	False	
y[0]	Real	0.0	False	True	True	False	
y[1]	Real	0.0	False	True	True	False	
y[2]	Real	0.0	False	True	True	False	
y[3]	Real	0.0	False	True	True	False	
y[4]	Real	0.0	False	True	True	False	
y[5]	Real	0.0	False	True	True	False	
test	USInt	0	False	True	True	False	
▼ u	Array[0..5] of Real		False	True	True	False	
u[0]	Real	0.0	False	True	True	False	
u[1]	Real	0.0	False	True	True	False	
u[2]	Real	0.0	False	True	True	False	
u[3]	Real	0.0	False	True	True	False	
u[4]	Real	0.0	False	True	True	False	
u[5]	Real	0.0	False	True	True	False	

test_FO_s1200 / PLC_1 [CPU 1212C AC/DC/Rly] / Program blocks

CFE [FB1]

CFE Properties

General

Name	CFE	Number	1	Type	FB	Language	SCL
------	-----	--------	---	------	----	----------	-----

Numbering automatic

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

CFE

Name	Data type	Default value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Input							
Uin	Real	1.0	Non-retain	True	True	False	sygnał wejściowy
▼ wm	Array[0..5] of Real		Retain	True	True	False	współczynniki licznika CFE (?)
wm[0]	Real	0.0	Retain	True	True	False	
wm[1]	Real	0.0	Retain	True	True	False	
wm[2]	Real	0.0	Retain	True	True	False	
wm[3]	Real	0.0	Retain	True	True	False	
wm[4]	Real	0.0	Retain	True	True	False	
wm[5]	Real	0.0	Retain	True	True	False	
▼ vm	Array[0..5] of Real		Retain	True	True	False	współczynniki mianownika CFE (?)
vm[0]	Real	0.0	Retain	True	True	False	
vm[1]	Real	0.0	Retain	True	True	False	
vm[2]	Real	0.0	Retain	True	True	False	
vm[3]	Real	0.0	Retain	True	True	False	
vm[4]	Real	0.0	Retain	True	True	False	
vm[5]	Real	0.0	Retain	True	True	False	
reset	Bool	false	Non-retain	True	True	False	
▼ Output							
licznik	Int	0	Non-retain	True	True	False	
Yout	Real	0.0	Non-retain	True	True	False	sygnał wyjściowy
InOut							
▼ Static							
▼ y	Array[0..5] of Real		Non-retain	True	True	False	
y[0]	Real	0.0	Non-retain	True	True	False	
y[1]	Real	0.0	Non-retain	True	True	False	
y[2]	Real	0.0	Non-retain	True	True	False	
y[3]	Real	0.0	Non-retain	True	True	False	
y[4]	Real	0.0	Non-retain	True	True	False	
y[5]	Real	0.0	Non-retain	True	True	False	
test	USInt	0	Non-retain	True	True	False	
▼ u	Array[0..5] of Real		Non-retain	True	True	False	
u[0]	Real	0.0	Non-retain	True	True	False	
u[1]	Real	0.0	Non-retain	True	True	False	
u[2]	Real	0.0	Non-retain	True	True	False	
u[3]	Real	0.0	Non-retain	True	True	False	
u[4]	Real	0.0	Non-retain	True	True	False	
u[5]	Real	0.0	Non-retain	True	True	False	
▼ Temp							
wsp_0	Real						
n	Int						
Constant							

```

0001 //reset wektorów przy zmianie współczynników modelu (aktywowany zmienną M0.0 "oblicz parametry")
0002 IF #reset THEN
0003   FOR #n:=0 TO 5 DO
0004     #y[#n] := 0;
0005     #u[#n] := 0;
0006   END_FOR;
0007   #licznik := 0;
0008 END_IF;
0009 //inicjalizacja sterowania:
0010 #u[5] := #Uin;
0011 //Aktualizacja licznika dla podanego skoku:
0012 IF #Uin > 0 THEN
0013   #licznik := #licznik + 1;
0014 END_IF;
0015 //obliczenia:
0016 //
0017 #wsp_0 := 1 / #vm[0];
0018 #y[5] := #wsp_0 * (-#vm[1] * #y[4] - #vm[2] * #y[3] - #vm[3] * #y[2] - #vm[4] * #y[1] - #vm[5] * #y[0]
+ #wm[0] * #u[5] + #wm[1] * #u[4] + #wm[2] * #u[3] + #wm[3] * #u[2] + #wm[4] * #u[1] + #wm[5] * #u[0]);
0019 //wynik:
0020 #Yout := 10.0 * #y[5];
0021 //przesunięcie wektorów wyjść:
0022 #y[0] := #y[1];

```

```

0023 #y[1] := #y[2];
0024 #y[2] := #y[3];
0025 #y[3] := #y[4];
0026 #y[4] := #y[5];
0027 //przesunięcie wektorów sterowań:
0028 #u[0] := #u[1];
0029 #u[1] := #u[2];
0030 #u[2] := #u[3];
0031 #u[3] := #u[4];
0032 #u[4] := #u[5];
0033 //test działania bloku:
0034 IF #test < 10 THEN
0035     #test := #test + 1;
0036 ELSE
0037     #test := 1;
0038 END_IF;
0039
0040 //
    
```

Symbol	Address	Type	Comment
#licznik		Int	
#n		Int	
#reset		Bool	
#test		USInt	
#u		Array	
#Uin		Real	sygnał wejściowy
#vm		Array	współczynniki mianownika CFE (?)
#wm		Array	współczynniki licznika CFE (?)
#wsp_0		Real	
#y		Array	
#Yout		Real	sygnał wyjściowy

test_FO_s1200 / PLC_1 [CPU 1212C AC/DC/Rly] / Program blocks

CFE_calc [FC1]

CFE_calc Properties

General

Name	CFE_calc	Number	1	Type	FC	Language	SCL
Numbering	automatic						

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

CFE_calc

Name	Data type	Default value	Comment
▼ Input			
n	USInt		rzad aproksymacji
T	Real		okres probkowania do obliczen
a	Real		współczynnik
r	Real		rzad ulamkowy do zamodelowania
▼ Output			
▼ CFE_licz	Array[0..5] of Real		
CFE_licz[0]	Real		
CFE_licz[1]	Real		
CFE_licz[2]	Real		
CFE_licz[3]	Real		
CFE_licz[4]	Real		
CFE_licz[5]	Real		
▼ CFE_mian	Array[0..5] of Real		
CFE_mian[0]	Real		
CFE_mian[1]	Real		
CFE_mian[2]	Real		
CFE_mian[3]	Real		
CFE_mian[4]	Real		
CFE_mian[5]	Real		
InOut			
▼ Temp			
k	Real		współczynnik
b0	Real		
b1	Real		
b2	Real		
b3	Real		
b4	Real		
b5	Real		
a0	Real		
a1	Real		
a2	Real		
a3	Real		
a4	Real		
a5	Real		
m	Int		wskaznik petli do inicjalizacji
Constant			
▼ Return			
CFE_calc	Void		

```

0001
0002
0003 //Funkcja różniczkowania i całkowania ułamkowego ze strony
0004 //Petrasa:
0005 //
0006 //http://people.tuke.sk/igor.podlubny/USU/matlab/petras/dfod1.m
0007 //function sysdfod=dfod1(n,T,a,r)
0008 // sysdfod=dfod1(n,T,a,r): new digital fractional order differentiator
0009 // AND integrator
0010 //
0011 // Output: :=>
0012 // Discrete system in the form OF the rational approximation - IIR filter
0013 // obtained BY continued fraction expansion OF a new generating function.
0014 //
0015 // Inputs: <:=
0016 // n: order OF truncation n:=(1 2 3 4 5)
0017 //
0018 //T: sampling period in [sec]
0019 // a: weighting factor between Euler AND Tustin rules ( 0 <:= a <:= 1 )
0020 // 0 - Euler rule, 1 - Tustin rule, 1/7 - Al-Alaoui rule, etc.
0021 // r: approximated fractional order (s**r), r is generally real number
0022 //
0023 // Author: Dr. Ivo Petras (ivo.petras@mail.com)
0024 // URL: http://ivopetras.tripod.com/
0025 //
0026 // Note: This approach is based on the original Al-Alaoui's work.
0027 //
0028 // Copyright (c), 2003.
0029 //

```

Totally Integrated Automation Portal		
<pre> 0030 //inicjalizacja: 0031 FOR #m := 0 TO 5 DO 0032 #CFE_licz[#m] := 0.0; 0033 #CFE_mian[#m] := 0.0; 0034 END_FOR; 0035 0036 0037 0038 #k := ((1+#a)/#T)**#r; 0039 0040 IF (#n=1) THEN 0041 #b0 := 2 / (#a * #r + #a + #r - 1); 0042 #b1 := - (#r + #a * #r - #a + 1) / (#a * #r + #a + #r - 1); 0043 #a0 := 2 / (#a * #r + #a + #r - 1); 0044 #a1 := 1; 0045 //przypisanie wyniku: 0046 #CFE_licz[0] := #b0; 0047 #CFE_licz[1] := #b1; 0048 // 0049 #CFE_mian[0] := #a0; 0050 #CFE_mian[1] := #a1; 0051 //sys1:=tf(k*[b0 b1],[a0 a1],T,'Variable', 'z**(-1)'); 0052 //sysdfod:=sys1; 0053 ELSIF (#n=2) THEN 0054 #b0 := 12 / (#a ** 2 * #r ** 2 + 3 * #a ** 2 * #r + 2 * #a * #r ** 2 + 2 * #a ** 2 + #r ** 2 - 8 * #a - 3 * #r + 2); 0055 #b1 := (12 * #a - 6 * #r - 6 * #a * #r - 12) / (#a ** 2 * #r ** 2 + 3 * #a ** 2 * #r + 2 * #a * #r ** 2 + 2 * #a ** 2 + #r ** 2 - 8 * #a - 3 * #r + 2); 0056 #b2 := (#a ** 2 * #r ** 2 - 3 * #a ** 2 * #r + 2 * #a * #r ** 2 + #r ** 2 - 8 * #a + 3 * #r + 2 * #a ** 2 + 2) / (#a ** 2 * #r ** 2 + 3 * #a ** 2 * #r + 2 * #a * #r ** 2 + 2 * #a ** 2 + #r ** 2 - 8 * #a - 3 * #r + 2); 0057 #a0 := 12 / (#a ** 2 * #r ** 2 + 3 * #a ** 2 * #r + 2 * #a * #r ** 2 + 2 * #a ** 2 + #r ** 2 - 8 * #a - 3 * #r + 2); 0058 #a1 := (12 * #a + 6 * #r + 6 * #a * #r - 12) / (#a ** 2 * #r ** 2 + 3 * #a ** 2 * #r + 2 * #a * #r ** 2 + 2 * #a ** 2 + #r ** 2 - 8 * #a - 3 * #r + 2); 0059 #a2 := 1; 0060 //sys2:=tf(k*[b0 b1 b2],[a0 a1 a2],T,'Variable', 'z**(-1)'); 0061 //sysdfod:=sys2; 0062 //przypisanie wyniku: 0063 #CFE_licz[0] := #b0; 0064 #CFE_licz[1] := #b1; 0065 #CFE_licz[2] := #b2; 0066 // 0067 #CFE_mian[0] := #a0; 0068 #CFE_mian[1] := #a1; 0069 #CFE_mian[2] := #a2; 0070 ELSIF (#n=3) THEN 0071 #b0 := 120 / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0072 #b1 := - (60 * #r + 180 - 180 * #a + 60 * #a * #r) / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0073 #b2 := - (-72 - 72 * #a ** 2 - 24 * #a * #r ** 2 + 216 * #a - 60 * #r - 12 * #r ** 2 - 12 * #a ** 2 * #r ** 2 + 60 * #a ** 2 * #r) / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0074 #b3 := - (11 * #r + 6 * #r ** 2 + #r ** 3 + 6 * #a * #r ** 2 - 6 * #a ** 2 * #r ** 2 - 27 * #a * #r - 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + #a ** 3 * #r ** 3 + 3 * #a * #r ** 3 + 11 * #a ** 3 * #r - 27 * #a ** 2 * #r - 6 * #a ** 3 - 54 * #a + 6 + 54 * #a ** 2) / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0075 #a0 := 120 / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0076 #a1 := (60 * #r - 180 + 180 * #a + 60 * #a * #r) / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0077 #a2 := (72 + 72 * #a ** 2 + 24 * #a * #r ** 2 - 216 * #a - 60 * #r + 12 * #r ** 2 + 12 * #a ** 2 * #r ** 2 + 60 * #a ** 2 * #r) / (#a ** 3 * #r ** 3 + 6 * #a ** 3 * #r ** 2 + 3 * #a ** 2 * #r ** 3 + 11 * #a ** 3 * #r + 6 * #a ** 2 * #r ** 2 + 3 * #a * #r ** 3 + 6 * #a ** 3 - 27 * #a ** 2 * #r - 6 * #a * #r ** 2 + #r ** 3 - 54 * #a ** 2 - 27 * #a * #r - 6 * #r ** 2 + 54 * #a + 11 * #r - 6); 0078 #a3 := 1; 0079 //przypisanie wyniku: 0080 #CFE_licz[0] := #b0; 0081 #CFE_licz[1] := #b1; 0082 #CFE_licz[2] := #b2; 0083 #CFE_licz[3] := #b3; 0084 // 0085 #CFE_mian[0] := #a0; 0086 #CFE_mian[1] := #a1; 0087 #CFE_mian[2] := #a2; 0088 #CFE_mian[3] := #a3; 0089 //sys3:=tf(k*[b0 b1 b2 b3],[a0 a1 a2 a3],T,'Variable', 'z**(-1)'); 0090 //sysdfod:=sys3; 0091 ELSIF (#n=4) THEN 0092 #b0 := 1680 / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); 0093 #b1 := (3360 * #a - 840 * #r - 840 * #a * #r - 3360) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - </pre>		

Totally Integrated Automation Portal		
	<pre> 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); 0094 #b2 := (-5760 * #a + 2160 * #a ** 2 + 2160 + 360 * #a * #r ** 2 + 1260 * #r + 180 * #r ** 2 - 1260 * #a ** 2 * #r + 180 * #a ** 2 * #r ** 2) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); 0095 #b3 := (-520 * #r + 960 * #a ** 2 * #r - 480 - 60 * #a ** 2 * #r ** 3 + 480 * #a ** 3 - 60 * #a * #r ** 3 + 180 * #a ** 3 * #r ** 2 - 520 * #a ** 3 * #r + 180 * #a ** 2 * #r ** 2 + 2880 * #a - 180 * #r ** 2 - 180 * #a * #r ** 2 - 20 * #r ** 3 + 960 * #a * #r - 20 * #a ** 3 * #r ** 3 - 2880 * #a ** 2) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); 0096 #b4 := (10 * #r ** 3 - 40 * #a * #r ** 2 + #r ** 4 - 150 * #a ** 2 * #r ** 2 + 35 * #r ** 2 - 384 * #a ** 3 - 384 * #a + 50 * #r - 320 * #a * #r + 24 + 20 * #a * #r ** 3 + #a ** 4 * #r ** 4 - 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 - 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 - 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 864 * #a ** 2 + 24 * #a ** 4 + 320 * #a ** 3 * #r) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); 0097 #a0 := 1680 / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); 0098 #a1 := (3360 * #a + 840 * #r + 840 * #a * #r - 3360) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); 0099 #a2 := (-5760 * #a + 2160 * #a ** 2 + 2160 + 360 * #a * #r ** 2 - 1260 * #r + 180 * #r ** 2 + 1260 * #a ** 2 * #r + 180 * #a ** 2 * #r ** 2) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); 0100 #a3 := (520 * #r - 960 * #a ** 2 * #r - 480 + 60 * #a ** 2 * #r ** 3 + 480 * #a ** 3 + 60 * #a * #r ** 3 + 180 * #a ** 3 * #r ** 2 + 520 * #a ** 3 * #r + 180 * #a ** 2 * #r ** 2 + 2880 * #a - 180 * #r ** 2 - 180 * #a * #r ** 2 + 20 * #r ** 3 - 960 * #a * #r + 20 * #a ** 3 * #r ** 3 - 2880 * #a ** 2) / (#a ** 4 * #r ** 4 + 10 * #a ** 4 * #r ** 3 + 4 * #a ** 3 * #r ** 4 + 35 * #a ** 4 * #r ** 2 + 20 * #a ** 3 * #r ** 3 + 6 * #a ** 2 * #r ** 4 + 50 * #a ** 4 * #r - 40 * #a ** 3 * #r ** 2 + 4 * #a * #r ** 4 + 24 * #a ** 4 - 320 * #a ** 3 * #r - 150 * #a ** 2 * #r ** 2 - 20 * #a * #r ** 3 + #r ** 4 - 384 * #a ** 3 - 40 * #a * #r ** 2 - 10 * #r ** 3 + 864 * #a ** 2 + 320 * #a * #r + 35 * #r ** 2 - 384 * #a - 50 * #r + 24); 0101 #a4 := 1; 0102 //przypisanie wyniku: 0103 #CFE_licz[0] := #b0; 0104 #CFE_licz[1] := #b1; 0105 #CFE_licz[2] := #b2; 0106 #CFE_licz[3] := #b3; 0107 #CFE_licz[4] := #b4; 0108 // 0109 #CFE_mian[0] := #a0; 0110 #CFE_mian[1] := #a1; 0111 #CFE_mian[2] := #a2; 0112 #CFE_mian[3] := #a3; 0113 #CFE_mian[4] := #a4; 0114 //sys4:=tf(k*[b0 b1 b2 b3 b4],[a0 a1 a2 a3 a4],T,'Variable', 'z**-1'); 0115 //sysdfod=sys4; 0116 ELSIF (#n=5) THEN 0117 #b0 := 30240 / (#a ** 5 * #r ** 5 + 15 * #a ** 5 * #r ** 4 + 5 * #a ** 4 * #r ** 5 + 85 * #a ** 5 * #r ** 3 + 45 * #a ** 4 * #r ** 4 + 10 * #a ** 3 * #r ** 5 + 225 * #a ** 5 * #r ** 2 + 5 * #a ** 4 * #r ** 3 + 30 * #a ** 3 * #r ** 4 + 10 * #a ** 2 * #r ** 5 + 274 * #a ** 5 * #r - 1005 * #a ** 4 * #r ** 2 - 410 * #a ** 3 * #r ** 3 - 30 * #a ** 2 * #r ** 4 + 5 * #a * #r ** 5 + 120 * #a ** 5 - 3250 * #a ** 4 * #r - 1230 * #a ** 3 * #r ** 2 - 410 * #a ** 2 * #r ** 3 - 45 * #a * #r ** 4 + #r ** 5 - 3000 * #a ** 4 + 4000 * #a ** 3 * #r + 1230 * #a ** 2 * #r ** 2 + 5 * #a * #r ** 3 - 15 * #r ** 4 + 12000 * #a ** 3 + 4000 * #a ** 2 * #r + 1005 * #a * #r ** 2 + 85 * #r ** 3 - 12000 * #a ** 2 - 3250 * #a * #r - 225 * #r ** 2 + 3000 * #a + 274 * #r - 120); 0118 #b1 := - (-75600 * #a + 15120 * #r + 15120 * #a * #r + 75600) / (#a ** 5 * #r ** 5 + 15 * #a ** 5 * #r ** 4 + 5 * #a ** 4 * #r ** 5 + 85 * #a ** 5 * #r ** 3 + 45 * #a ** 4 * #r ** 4 + 10 * #a ** 3 * #r ** 5 + 225 * #a ** 5 * #r ** 2 + 5 * #a ** 4 * #r ** 3 + 30 * #a ** 3 * #r ** 4 + 10 * #a ** 2 * #r ** 5 + 274 * #a ** 5 * #r - 1005 * #a ** 4 * #r ** 2 - 410 * #a ** 3 * #r ** 3 - 30 * #a ** 2 * #r ** 4 + 5 * #a * #r ** 5 + 120 * #a ** 5 - 3250 * #a ** 4 * #r - 1230 * #a ** 3 * #r ** 2 - 410 * #a ** 2 * #r ** 3 - 45 * #a * #r ** 4 + #r ** 5 - 3000 * #a ** 4 + 4000 * #a ** 3 * #r + 1230 * #a ** 2 * #r ** 2 + 5 * #a * #r ** 3 - 15 * #r ** 4 + 12000 * #a ** 3 + 4000 * #a ** 2 * #r + 1005 * #a * #r ** 2 + 85 * #r ** 3 - 12000 * #a ** 2 - 3250 * #a * #r - 225 * #r ** 2 + 3000 * #a + 274 * #r - 120); 0119 #b2 := - (168000 * #a - 30240 * #r - 67200 - 3360 * #a ** 2 * #r ** 2 - 3360 * #r ** 2 - 67200 * #a ** 2 - 6720 * #a * #r ** 2 + 30240 * #a ** 2 * #r) / (#a ** 5 * #r ** 5 + 15 * #a ** 5 * #r ** 4 + 5 * #a ** 4 * #r ** 5 + 85 * #a ** 5 * #r ** 3 + 45 * #a ** 4 * #r ** 4 + 10 * #a ** 3 * #r ** 5 + 225 * #a ** 5 * #r ** 2 + 5 * #a ** 4 * #r ** 3 + 30 * #a ** 3 * #r ** 4 + 10 * #a ** 2 * #r ** 5 + 274 * #a ** 5 * #r - 1005 * #a ** 4 * #r ** 2 - 410 * #a ** 3 * #r ** 3 - 30 * #a ** 2 * #r ** 4 + 5 * #a * #r ** 5 + 120 * #a ** 5 - 3250 * #a ** 4 * #r - 1230 * #a ** 3 * #r ** 2 - 410 * #a ** 2 * #r ** 3 - 45 * #a * #r ** 4 + #r ** 5 - 3000 * #a ** 4 + 4000 * #a ** 3 * #r + 1230 * #a ** 2 * #r ** 2 + 5 * #a * #r ** 3 - 15 * #r ** 4 + 12000 * #a ** 3 + 4000 * #a ** 2 * #r + 1005 * #a * #r ** 2 + 85 * #r ** 3 - 12000 * #a ** 2 - 3250 * #a * #r - 225 * #r ** 2 + 3000 * #a + 274 * #r - 120); 0120 #b3 := - (5040 * #a * #r ** 2 - 31500 * #a ** 2 * #r - 5040 * #a ** 2 * #r ** 2 - 25200 * #a ** 3 + 19740 * #a ** 3 * #r + 5040 * #r ** 2 + 1260 * #a * #r ** 3 + 126000 * #a ** 2 - 31500 * #a * #r + 420 * #a ** 3 * #r ** 3 + 19740 * #r + 420 * #r ** 3 + 1260 * #a ** 2 * #r ** 3 + 25200 - 5040 * #a ** 3 * #r ** 2 - 126000 * #a) / (#a ** 5 * #r ** 5 + 15 * #a ** 5 * #r ** 4 + 5 * #a ** 4 * #r ** 5 + 85 * #a ** 5 * #r ** 3 + 45 * #a ** 4 * #r ** 4 + 10 * #a ** 3 * #r ** 5 + 225 * #a ** 5 * #r ** 2 + 5 * #a ** 4 * #r ** 3 + 30 * #a ** 3 * #r </pre>	


```

0142 #CFE_mian[5] := #a5;
0143 //sys5:=tf(k*[b0 b1 b2 b3 b4 b5],[a0 a1 a2 a3 a4 a5],T,'Variable', 'z**'-1');
0144 //sysdfod:=sys5;
0145 END_IF;
0146 //
0147
0148
0149
0150

```

Symbol	Address	Type	Comment
#a		Real	współczynnik
#a0		Real	
#a1		Real	
#a2		Real	
#a3		Real	
#a4		Real	
#a5		Real	
#b0		Real	
#b1		Real	
#b2		Real	
#b3		Real	
#b4		Real	
#b5		Real	
#CFE_licz		Array	
#CFE_mian		Array	
#k		Real	współczynnik
#m		Int	wskaznik petli do inicjalizacji
#n		USInt	rzad aproksymacji
#r		Real	rzad ulamkowy do zamodelowania
#T		Real	okres probkowania do obliczen

test_FO_s1200 / PLC_1 [CPU 1212C AC/DC/Rly] / Program blocks

Main [OB1]

Main Properties

General

Name	Main	Number	1	Type	OB	Language	LAD
-------------	------	---------------	---	-------------	----	-----------------	-----

Numbering automatic

Information

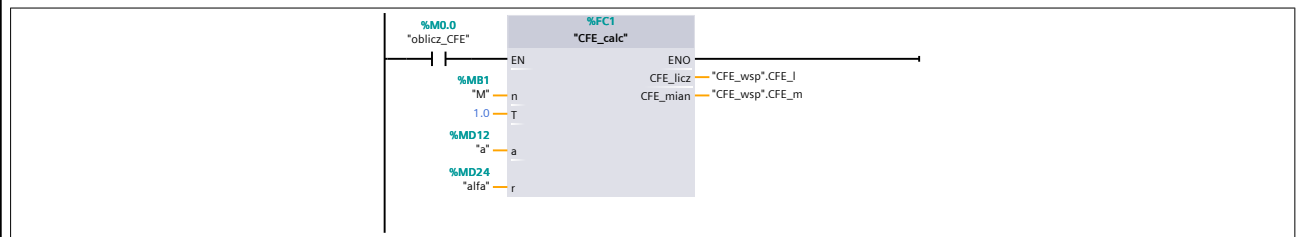
Title	program glowny	Author		Comment		Family	
Version	0.1	User-defined ID					

Main

Name	Data type	Default value	Comment
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
Temp			
Constant			

Network 1:

Obliczanie współczynników aproksymacji CFE: warunkowe, jeden raz po zmianie parametrów. Współczynniki są zapamiętywane a potem pobierane gotowe podczas obliczeń.



Symbol	Address	Type	Comment
"a"	%MD12	Real	
"alfa"	%MD24	Real	
"CFE_wsp".CFE_l		Array	
"CFE_wsp".CFE_m		Array	
"M"	%MB1	USInt	
"oblicz_CFE"	%M0.0	Bool	

test_FO_s1200 / PLC_1 [CPU 1212C AC/DC/Rly] / Program blocks

Cyclic interrupt [OB30]

Cyclic interrupt Properties

General

Name	Cyclic interrupt	Number	30	Type	OB	Language	LAD
Numbering	automatic						

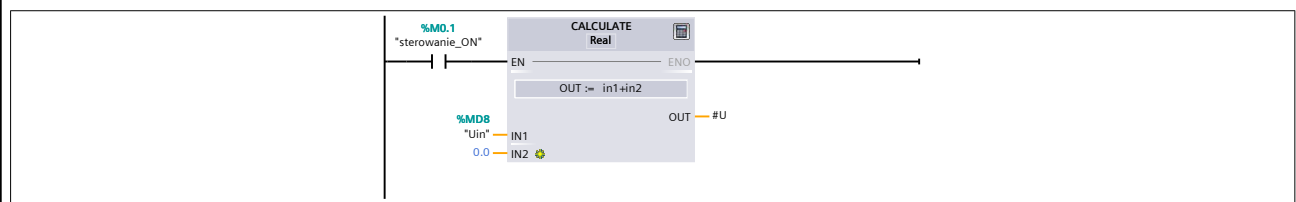
Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Cyclic interrupt

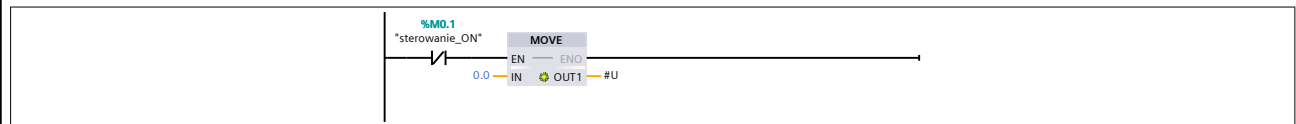
Name	Data type	Default value	Comment
▼ Input			
Initial_Call	Bool		Initial call of this OB
Event_Count	Int		Events discarded
▼ Temp			
U	Real		
Constant			

Network 1:



Symbol	Address	Type	Comment
"sterowanie_ON"	%M0.1	Bool	
"Uin"	%MD8	Real	
#U		Real	

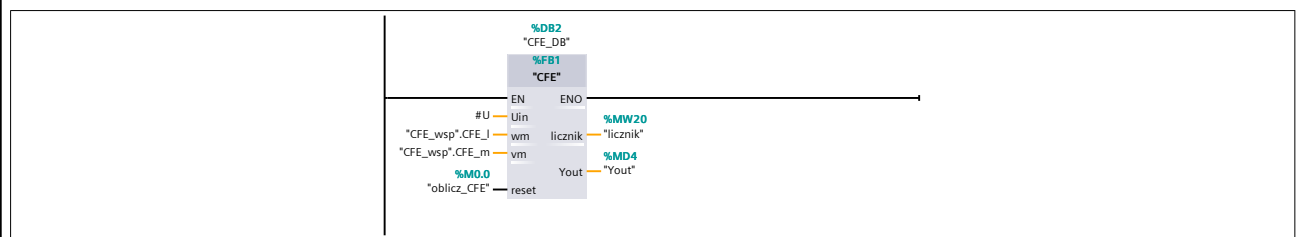
Network 2:



Symbol	Address	Type	Comment
"sterowanie_ON"	%M0.1	Bool	
#U		Real	

Network 3:









Wyznaczanie pochodnej lub całki ułamkowej z sygnału Uin. Parametry aproksymacji są pobierane z bloku danych CFE_wsp. Sygnał sterujący to Uin. Wyjście należy zebrać w aplikacji SCADA.



Symbol	Address	Type	Comment
"CFE_wsp".CFE_l		Array	
"CFE_wsp".CFE_m		Array	
"licznik"	%MW20	Int	
"oblicz_CFE"	%M0.0	Bool	
"Yout"	%MD4	Real	
#U		Real	

test_FO_s1200 / PLC_1 [CPU 1212C AC/DC/Rly]

PLC tags

	Name	Data type	Address	Retain	Visible in HMI	Accessible from HMI	Comment
	oblicz_CFE	Bool	%M0.0	False	True	True	
	Yout	Real	%MD4	False	True	True	
	Uin	Real	%MD8	False	True	True	
	M	USInt	%MB1	False	True	True	
	a	Real	%MD12	False	True	True	
	alfa	Real	%MD24	False	True	True	
	sterowanie_ON	Bool	%M0.1	False	True	True	
	licznik	Int	%MW20	False	True	True	

test_FO_s1200 / PSE [CPU 1511-1 PN] / Program blocks / 4 Math operations

LGF_AverageAndDeviation [FC10006]

LGF_AverageAndDeviation Properties

General

Name	LGF_AverageAndDeviation	Number	10006	Type	FC	Language	SCL
------	-------------------------	--------	-------	------	----	----------	-----

Numbering automatic

Information

Title		Author		Comment		Family	
Version		User-defined ID					

LGF_AverageAndDeviation

Name	Data type	Default value	Comment
▼ Input			
variableArray	Variant		
▼ Output			
arithmeticAverage	Real		
standardDeviation	Real		
error	Bool		
statusID	UInt		
status	Word		
InOut			
▼ Temp			
tempArraySize	UDInt		
tempSizeCounter	Int		
tempInternalError	Int		
tempActValueInt	Int		
tempActValueUInt	UInt		
tempActValueDInt	DInt		
tempActValueUDInt	UDInt		
tempActValueUSInt	USInt		
tempActValueSInt	SInt		
tempActValueReal	Real		
tempArithmeticAverage	Real		
tempStandardDeviation	Real		
▼ Constant			
NO_ERROR	Word	16#0000	
NO_CURRENT_JOBS	Word	16#7000	
NO_ARRAY	Word	16#8200	
WRONG_TYPE	Word	16#8201	
ERROR_IN_THIS_BLOCK	UInt	1	
ERROR_MOVE_BLK_VARIANT	UInt	2	
▼ Return			
LGF_AverageAndDeviation	Void		

```

0001 //=====
0002 // SIEMENS AG
0003 // (c)Copyright (2015)
0004 //-----
0005 // Library: LGF (Library General Functions)
0006 // Tested with: CPU1212C DC/DC/DC FW:V4.1
0007 // Engineering: TIA Portal V13 SP1 Upd 5
0008 // Restrictions: -
0009 // Requirements: PLC (S7-1200 / S7-1500)
0010 // Functionality: This function calculates the arithmetic average and the
0011 // standard Deviation of a array of numbers
0012 //-----
0013 // Change log table:
0014 // Version Date In charge Changes applied
0015 // 01.00.00 19.08.2015 Siemens Industry Online Support First released version
0016 // 01.00.01 16.11.2015 Siemens Industry Online Support
0017 // Bug fix at WRONG_TYPE: #error := true
0018 //=====
0019
0020 //set "No current job" status
0021 #error := false;
0022 #statusID := #ERROR_IN_THIS_BLOCK;
0023 #status := #NO_CURRENT_JOBS;
0024
0025 // Check if the variant-variable is an array if yes cout the elements of the array
0026 IF NOT (IS_ARRAY(#variableArray)) THEN
0027     #error := true;
0028     #statusID := #ERROR_IN_THIS_BLOCK;
0029     #status := #NO_ARRAY;
0030     RETURN;
0031 ELSE
0032     #tempArraySize := CountOfElements(#variableArray);
0033 END_IF;
0034
0035 CASE TypeOfElements(#variableArray) OF
0036     Int:
0037     // Calculation of the arithmetic average
0038     #tempArithmeticAverage := 0;

```

Totally Integrated Automation Portal		
<pre> 0039 FOR #tempSizeCounter := 0 TO UDINT_TO_INT(#tempArraySize) - 1 DO 0040 #tempInternalError := MOVE_BLK_VARIANT(SRC := #variableArray, 0041 COUNT := 1, 0042 SRC_INDEX := #tempSizeCounter, 0043 DEST_INDEX := 0, 0044 DEST => #tempActValueInt); 0045 0046 #tempArithmeticAverage := #tempArithmeticAverage + INT_TO_REAL(#tempActValueInt); 0047 END_FOR; 0048 0049 #tempArithmeticAverage := #tempArithmeticAverage / UDINT_TO_REAL(#tempArraySize); 0050 0051 //----- 0052 //Calculation of the standardDeviation 0053 #tempStandardDeviation := 0; 0054 FOR #tempSizeCounter := 0 TO UDINT_TO_INT(#tempArraySize) - 1 DO 0055 #tempInternalError := MOVE_BLK_VARIANT(SRC := #variableArray, 0056 COUNT := 1, 0057 SRC_INDEX := #tempSizeCounter, 0058 DEST_INDEX := 0, 0059 DEST => #tempActValueInt); 0060 0061 #tempStandardDeviation := #tempStandardDeviation + SQR((INT_TO_REAL(#tempActValueInt) - #tempArithmeticA- 0062 verage)); 0063 END_FOR; 0064 #tempStandardDeviation := #tempStandardDeviation / UDINT_TO_REAL(#tempArraySize); 0065 #tempStandardDeviation := SQR(#tempStandardDeviation); 0066 0067 //----- 0068 DInt: 0069 // Calculation of the arithmetic average 0070 #tempArithmeticAverage := 0; 0071 FOR #tempSizeCounter := 0 TO UDINT_TO_INT(#tempArraySize) - 1 DO 0072 #tempInternalError := MOVE_BLK_VARIANT(SRC := #variableArray, 0073 COUNT := 1, 0074 SRC_INDEX := #tempSizeCounter, 0075 DEST_INDEX := 0, 0076 DEST => #tempActValueDInt); 0077 0078 #tempArithmeticAverage := #tempArithmeticAverage + DINT_TO_REAL(#tempActValueDInt); 0079 END_FOR; 0080 0081 #tempArithmeticAverage := #tempArithmeticAverage / UDINT_TO_REAL(#tempArraySize); 0082 0083 //----- 0084 //Calculation of the standardDeviation 0085 #tempStandardDeviation := 0; 0086 FOR #tempSizeCounter := 0 TO UDINT_TO_INT(#tempArraySize) - 1 DO 0087 #tempInternalError := MOVE_BLK_VARIANT(SRC := #variableArray, 0088 COUNT := 1, 0089 SRC_INDEX := #tempSizeCounter, 0090 DEST_INDEX := 0, 0091 DEST => #tempActValueDInt); 0092 0093 #tempStandardDeviation := #tempStandardDeviation + SQR((DINT_TO_REAL(#tempActValueDInt) - #tempArithmeticA- 0094 verage)); 0095 END_FOR; 0096 #tempStandardDeviation := #tempStandardDeviation / UDINT_TO_REAL(#tempArraySize); 0097 #tempStandardDeviation := SQR(#tempStandardDeviation); 0098 0099 //----- 0100 UInt: 0101 // Calculation of the arithmetic average 0102 #tempArithmeticAverage := 0; 0103 FOR #tempSizeCounter := 0 TO UDINT_TO_INT(#tempArraySize) - 1 DO 0104 #tempInternalError := MOVE_BLK_VARIANT(SRC := #variableArray, 0105 COUNT := 1, 0106 SRC_INDEX := #tempSizeCounter, 0107 DEST_INDEX := 0, 0108 DEST => #tempActValueUInt); 0109 0110 #tempArithmeticAverage := #tempArithmeticAverage + UINT_TO_REAL(#tempActValueUInt); 0111 END_FOR; 0112 0113 #tempArithmeticAverage := #tempArithmeticAverage / UDINT_TO_REAL(#tempArraySize); 0114 0115 //----- 0116 //Calculation of the standardDeviation 0117 #tempStandardDeviation := 0; 0118 FOR #tempSizeCounter := 0 TO UDINT_TO_INT(#tempArraySize) - 1 DO 0119 #tempInternalError := MOVE_BLK_VARIANT(SRC := #variableArray, 0120 COUNT := 1, 0121 SRC_INDEX := #tempSizeCounter, 0122 DEST_INDEX := 0, 0123 DEST => #tempActValueUInt); 0124 0125 #tempStandardDeviation := #tempStandardDeviation + SQR((UINT_TO_REAL(#tempActValueUInt) - #tempArithmeticA- 0126 verage)); 0127 END_FOR; </pre>		

Totally Integrated Automation Portal		
<pre> 0127 0128 #tempStandardDeviation := #tempStandardDeviation / UDINT_TO_REAL(#tempArraySize); 0129 #tempStandardDeviation := SQRT(#tempStandardDeviation); 0130 0131 //----- 0132 UDInt: 0133 // Calculation of the arithmetic average 0134 #tempArithmeticAverage := 0; 0135 FOR #tempSizeCounter := 0 TO UDINT_TO_INT(#tempArraySize) - 1 DO 0136 #tempInternalError := MOVE_BLK_VARIANT(SRC := #variableArray, 0137 COUNT := 1, 0138 SRC_INDEX := #tempSizeCounter, 0139 DEST_INDEX := 0, 0140 DEST => #tempActValueUDInt); 0141 0142 #tempArithmeticAverage := #tempArithmeticAverage + UDINT_TO_REAL(#tempActValueUDInt); 0143 END_FOR; 0144 0145 #tempArithmeticAverage := #tempArithmeticAverage / UDINT_TO_REAL(#tempArraySize); 0146 0147 //----- 0148 //Calculation of the standardDeviation 0149 #tempStandardDeviation := 0; 0150 FOR #tempSizeCounter := 0 TO UDINT_TO_INT(#tempArraySize) - 1 DO 0151 #tempInternalError := MOVE_BLK_VARIANT(SRC := #variableArray, 0152 COUNT := 1, 0153 SRC_INDEX := #tempSizeCounter, 0154 DEST_INDEX := 0, 0155 DEST => #tempActValueUDInt); 0156 0157 #tempStandardDeviation := #tempStandardDeviation + SQR((UDINT_TO_REAL(#tempActValueUDInt) - #tempArithmeti- cAverage)); 0158 END_FOR; 0159 0160 #tempStandardDeviation := #tempStandardDeviation / UDINT_TO_REAL(#tempArraySize); 0161 #tempStandardDeviation := SQRT(#tempStandardDeviation); 0162 0163 //-----; 0164 USInt: 0165 // Calculation of the arithmetic average 0166 #tempArithmeticAverage := 0; 0167 FOR #tempSizeCounter := 0 TO UDINT_TO_INT(#tempArraySize) - 1 DO 0168 #tempInternalError := MOVE_BLK_VARIANT(SRC := #variableArray, 0169 COUNT := 1, 0170 SRC_INDEX := #tempSizeCounter, 0171 DEST_INDEX := 0, 0172 DEST => #tempActValueUSInt); 0173 0174 #tempArithmeticAverage := #tempArithmeticAverage + USINT_TO_REAL(#tempActValueUSInt); 0175 END_FOR; 0176 0177 #tempArithmeticAverage := #tempArithmeticAverage/ UDINT_TO_REAL(#tempArraySize); 0178 0179 //----- 0180 //Calculation of the standardDeviation 0181 #tempStandardDeviation := 0; 0182 FOR #tempSizeCounter := 0 TO UDINT_TO_INT(#tempArraySize) - 1 DO 0183 #tempInternalError := MOVE_BLK_VARIANT(SRC := #variableArray, 0184 COUNT := 1, 0185 SRC_INDEX := #tempSizeCounter, 0186 DEST_INDEX := 0, 0187 DEST => #tempActValueUSInt); 0188 0189 #tempStandardDeviation := #tempStandardDeviation + SQR((USINT_TO_REAL(#tempActValueUSInt) - #tempArithmeti- cAverage)); 0190 END_FOR; 0191 0192 #tempStandardDeviation := #tempStandardDeviation / UDINT_TO_REAL(#tempArraySize); 0193 #tempStandardDeviation := SQRT(#tempStandardDeviation); 0194 0195 //----- 0196 SInt: 0197 // Calculation of the arithmetic average 0198 #tempArithmeticAverage := 0; 0199 FOR #tempSizeCounter := 0 TO UDINT_TO_INT(#tempArraySize) - 1 DO 0200 #tempInternalError := MOVE_BLK_VARIANT(SRC := #variableArray, 0201 COUNT := 1, 0202 SRC_INDEX := #tempSizeCounter, 0203 DEST_INDEX := 0, 0204 DEST => #tempActValueSInt); 0205 0206 #tempArithmeticAverage := #tempArithmeticAverage + SINT_TO_REAL(#tempActValueSInt); 0207 END_FOR; 0208 0209 #tempArithmeticAverage := #tempArithmeticAverage / UDINT_TO_REAL(#tempArraySize); 0210 0211 //----- 0212 //Calculation of the standardDeviation 0213 #tempStandardDeviation := 0; 0214 FOR #tempSizeCounter := 0 TO UDINT_TO_INT(#tempArraySize) - 1 DO 0215 #tempInternalError := MOVE_BLK_VARIANT(SRC := #variableArray, </pre>		


```

0216             COUNT := 1,
0217             SRC_INDEX := #tempSizeCounter,
0218             DEST_INDEX := 0,
0219             DEST => #tempActValueSInt);
0220
0221     #tempStandardDeviation := #tempStandardDeviation + SQR((SINT_TO_REAL(#tempActValueSInt) - #tempArithmeticA-
verage));
0222 END_FOR;
0223
0224 #tempStandardDeviation := #tempStandardDeviation / UDINT_TO_REAL(#tempArraySize);
0225 #tempStandardDeviation := SQRT(#tempStandardDeviation);
0226
0227 //-----
0228 Real:
0229 // Calculation of the arithmetic average
0230 #tempArithmeticAverage := 0;
0231 FOR #tempSizeCounter := 0 TO UDINT_TO_INT(#tempArraySize) - 1 DO
0232     #tempInternalError := MOVE_BLK_VARIANT(SRC := #variableArray,
0233     COUNT := 1,
0234     SRC_INDEX := #tempSizeCounter,
0235     DEST_INDEX := 0,
0236     DEST => #tempActValueReal);
0237
0238     #tempArithmeticAverage := #tempArithmeticAverage + #tempActValueReal;
0239 END_FOR;
0240
0241 #tempArithmeticAverage := #tempArithmeticAverage / UDINT_TO_REAL(#tempArraySize);
0242
0243 //-----
0244 //Calculation of the standardDeviation
0245 #tempStandardDeviation := 0;
0246 FOR #tempSizeCounter := 0 TO UDINT_TO_INT(#tempArraySize) - 1 DO
0247     #tempInternalError := MOVE_BLK_VARIANT(SRC := #variableArray,
0248     COUNT := 1,
0249     SRC_INDEX := #tempSizeCounter,
0250     DEST_INDEX := 0,
0251     DEST => #tempActValueReal);
0252
0253     #tempStandardDeviation := #tempStandardDeviation + SQR((#tempActValueReal - #tempArithmeticAverage));
0254 END_FOR;
0255
0256 #tempStandardDeviation := #tempStandardDeviation / UDINT_TO_REAL(#tempArraySize);
0257 #tempStandardDeviation := SQRT(#tempStandardDeviation);
0258
0259 //-----
0260 ELSE
0261     #error := true;
0262     #statusID := #ERROR_IN_THIS_BLOCK;
0263     #status := #WRONG_TYPE;
0264     RETURN;
0265 END_CASE;
0266 #arithmeticAverage := #tempArithmeticAverage;
0267 #standardDeviation := #tempStandardDeviation;
0268
0269 IF #tempInternalError > 1 THEN
0270     #error := true;
0271     #statusID := #ERROR_MOVE_BLK_VARIANT;
0272     #status := INT_TO_WORD(#tempInternalError);
0273     RETURN;
0274 END_IF;
0275
0276 #status := #NO_ERROR;

```

Symbol	Address	Type	Comment
#arithmeticAverage		Real	
#error		Bool	
#ERROR_IN_THIS_BLOCK	1	UInt	
#ERROR_MOVE_BLK_VARIANT	2	UInt	
#NO_ARRAY	16#8200	Word	
#NO_CURRENT_JOBS	16#7000	Word	
#NO_ERROR	16#0000	Word	
#standardDeviation		Real	
#status		Word	
#statusID		UInt	
#tempActValueDInt		DInt	
#tempActValueInt		Int	
#tempActValueReal		Real	
#tempActValueSInt		SInt	
#tempActValueUDInt		UDInt	
#tempActValueUInt		UInt	
#tempActValueUSInt		USInt	
#tempArithmeticAverage		Real	
#tempArraySize		UDInt	
#tempInternalError		Int	
#tempSizeCounter		Int	
#tempStandardDeviation		Real	
#variableArray		VARIANT	
#WRONG_TYPE	16#8201	Word	

test_FO_s1200 / PSE [CPU 1511-1 PN] / Program blocks

Main [OB1]

Main Properties

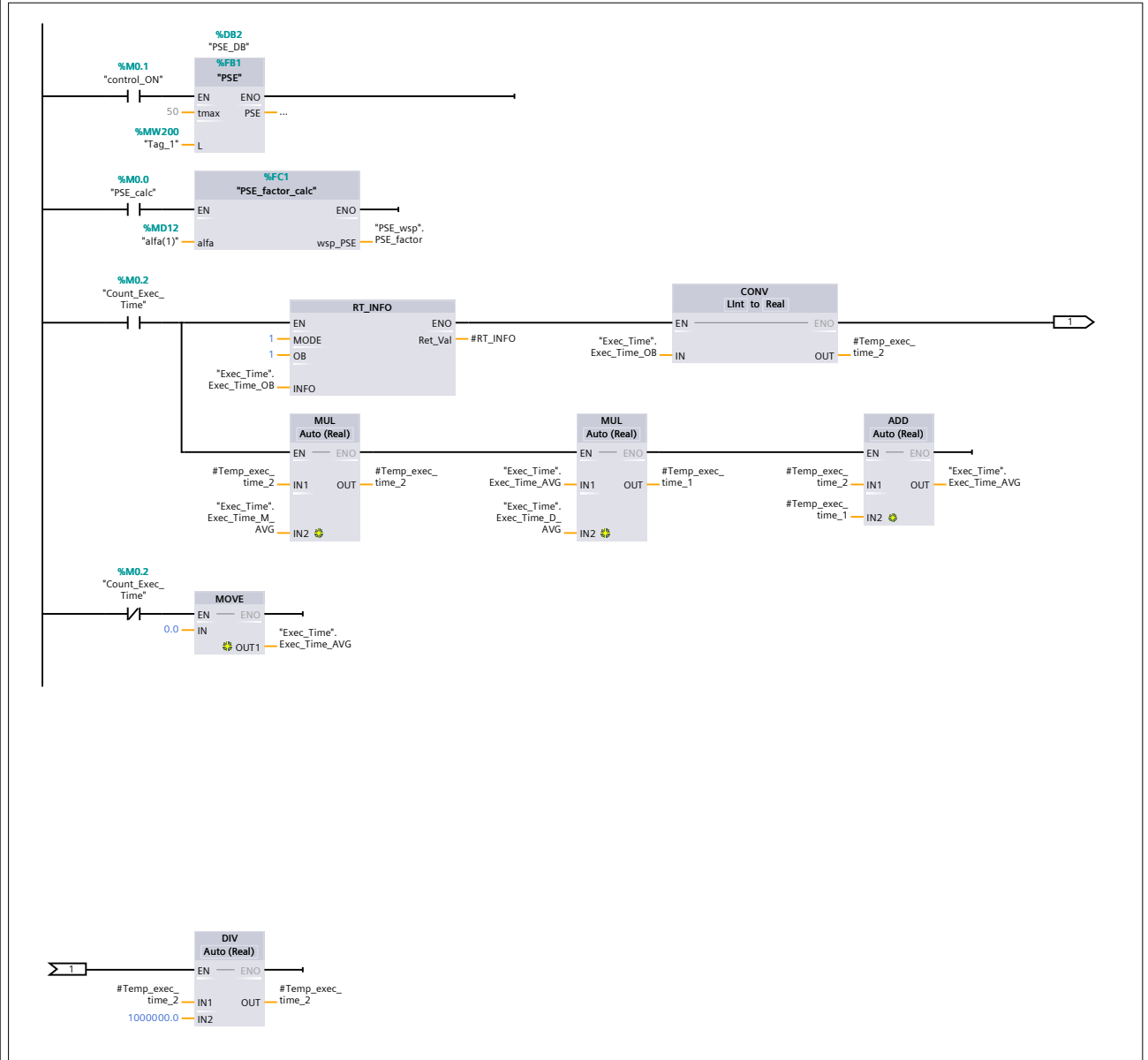
General							
Name	Main	Number	1	Type	OB	Language	LAD
Numbering	automatic						
Information							
Title	main	Author		Comment		Family	
Version	0.1	User-defined ID					

Main

Name	Data type	Default value	Comment
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
▼ Temp			
RT_INFO	Int		
Temp_exec_time_1	Real		
Temp_exec_time_2	Real		
Dif_actual_avg_time	Real		
Constant			

Network 1:

Network 1:



Symbol	Address	Type	Comment
"alfa(1)"	%MD12	Real	
"control_ON"	%M0.1	Bool	

Symbol	Address	Type	Comment
"Count_Exec_Time"	%M0.2	Bool	
"Exec_Time".Exec_Time_AVG		Real	
"Exec_Time".Exec_Time_D_AVG		Real	
"Exec_Time".Exec_Time_M_AVG		Real	
"Exec_Time".Exec_Time_OB		LTime	
"PSE_calc"	%M0.0	Bool	
"PSE_wsp".PSE_factor		Array	
"Tag_1"	%MW200	Int	
#RT_INFO		Int	
#Temp_exec_time_1		Real	
#Temp_exec_time_2		Real	

test_FO_s1200 / PSE [CPU 1511-1 PN] / Program blocks

PSE_DB [DB2]

PSE_DB Properties

General

Name	PSE_DB	Number	2	Type	DB	Language	DB
-------------	--------	---------------	---	-------------	----	-----------------	----

Numbering	automatic
------------------	-----------

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

PSE_DB

Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Input							
tmax	Int	50	False	True	True	False	
L	Int	10	False	True	True	False	
▼ Output							
▼ PSE	Array[1..300] of Real		False	True	True	False	
PSE[1]	Real	0.0	False	True	True	False	
PSE[2]	Real	0.0	False	True	True	False	
PSE[3]	Real	0.0	False	True	True	False	
PSE[4]	Real	0.0	False	True	True	False	
PSE[5]	Real	0.0	False	True	True	False	
PSE[6]	Real	0.0	False	True	True	False	
PSE[7]	Real	0.0	False	True	True	False	
PSE[8]	Real	0.0	False	True	True	False	
PSE[9]	Real	0.0	False	True	True	False	
PSE[10]	Real	0.0	False	True	True	False	
PSE[11]	Real	0.0	False	True	True	False	
PSE[12]	Real	0.0	False	True	True	False	
PSE[13]	Real	0.0	False	True	True	False	
PSE[14]	Real	0.0	False	True	True	False	
PSE[15]	Real	0.0	False	True	True	False	
PSE[16]	Real	0.0	False	True	True	False	
PSE[17]	Real	0.0	False	True	True	False	
PSE[18]	Real	0.0	False	True	True	False	
PSE[19]	Real	0.0	False	True	True	False	
PSE[20]	Real	0.0	False	True	True	False	
PSE[21]	Real	0.0	False	True	True	False	
PSE[22]	Real	0.0	False	True	True	False	
PSE[23]	Real	0.0	False	True	True	False	
PSE[24]	Real	0.0	False	True	True	False	
PSE[25]	Real	0.0	False	True	True	False	
PSE[26]	Real	0.0	False	True	True	False	
PSE[27]	Real	0.0	False	True	True	False	
PSE[28]	Real	0.0	False	True	True	False	
PSE[29]	Real	0.0	False	True	True	False	
PSE[30]	Real	0.0	False	True	True	False	
PSE[31]	Real	0.0	False	True	True	False	
PSE[32]	Real	0.0	False	True	True	False	
PSE[33]	Real	0.0	False	True	True	False	
PSE[34]	Real	0.0	False	True	True	False	
PSE[35]	Real	0.0	False	True	True	False	
PSE[36]	Real	0.0	False	True	True	False	
PSE[37]	Real	0.0	False	True	True	False	
PSE[38]	Real	0.0	False	True	True	False	
PSE[39]	Real	0.0	False	True	True	False	
PSE[40]	Real	0.0	False	True	True	False	
PSE[41]	Real	0.0	False	True	True	False	
PSE[42]	Real	0.0	False	True	True	False	
PSE[43]	Real	0.0	False	True	True	False	
PSE[44]	Real	0.0	False	True	True	False	
PSE[45]	Real	0.0	False	True	True	False	
PSE[46]	Real	0.0	False	True	True	False	
PSE[47]	Real	0.0	False	True	True	False	
PSE[48]	Real	0.0	False	True	True	False	
PSE[49]	Real	0.0	False	True	True	False	
PSE[50]	Real	0.0	False	True	True	False	
PSE[51]	Real	0.0	False	True	True	False	
PSE[52]	Real	0.0	False	True	True	False	
PSE[53]	Real	0.0	False	True	True	False	
PSE[54]	Real	0.0	False	True	True	False	
PSE[55]	Real	0.0	False	True	True	False	
PSE[56]	Real	0.0	False	True	True	False	
PSE[57]	Real	0.0	False	True	True	False	
PSE[58]	Real	0.0	False	True	True	False	
PSE[59]	Real	0.0	False	True	True	False	
PSE[60]	Real	0.0	False	True	True	False	

Totally Integrated Automation Portal								
Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment	
InOut								
▼ Static								
▼ U	Array[1..300] of Real		True	True	True	False		
U[1]	Real	10.0	True	True	True	False		
U[2]	Real	10.0	True	True	True	False		
U[3]	Real	10.0	True	True	True	False		
U[4]	Real	10.0	True	True	True	False		
U[5]	Real	10.0	True	True	True	False		
U[6]	Real	10.0	True	True	True	False		
U[7]	Real	10.0	True	True	True	False		
U[8]	Real	10.0	True	True	True	False		
U[9]	Real	10.0	True	True	True	False		
U[10]	Real	10.0	True	True	True	False		
U[11]	Real	10.0	True	True	True	False		
U[12]	Real	10.0	True	True	True	False		
U[13]	Real	10.0	True	True	True	False		
U[14]	Real	10.0	True	True	True	False		
U[15]	Real	10.0	True	True	True	False		
U[16]	Real	10.0	True	True	True	False		
U[17]	Real	10.0	True	True	True	False		
U[18]	Real	10.0	True	True	True	False		
U[19]	Real	10.0	True	True	True	False		
U[20]	Real	10.0	True	True	True	False		
U[21]	Real	10.0	True	True	True	False		
U[22]	Real	10.0	True	True	True	False		
U[23]	Real	10.0	True	True	True	False		
U[24]	Real	10.0	True	True	True	False		
U[25]	Real	10.0	True	True	True	False		
U[26]	Real	10.0	True	True	True	False		
U[27]	Real	10.0	True	True	True	False		
U[28]	Real	10.0	True	True	True	False		
U[29]	Real	10.0	True	True	True	False		
U[30]	Real	10.0	True	True	True	False		
U[31]	Real	10.0	True	True	True	False		
U[32]	Real	10.0	True	True	True	False		
U[33]	Real	10.0	True	True	True	False		
U[34]	Real	10.0	True	True	True	False		
U[35]	Real	10.0	True	True	True	False		
U[36]	Real	10.0	True	True	True	False		
U[37]	Real	10.0	True	True	True	False		
U[38]	Real	10.0	True	True	True	False		
U[39]	Real	10.0	True	True	True	False		
U[40]	Real	10.0	True	True	True	False		
U[41]	Real	10.0	True	True	True	False		
U[42]	Real	10.0	True	True	True	False		
U[43]	Real	10.0	True	True	True	False		
U[44]	Real	10.0	True	True	True	False		
U[45]	Real	10.0	True	True	True	False		
U[46]	Real	10.0	True	True	True	False		
U[47]	Real	10.0	True	True	True	False		
U[48]	Real	10.0	True	True	True	False		
U[49]	Real	10.0	True	True	True	False		
U[50]	Real	10.0	True	True	True	False		
U[51]	Real	0.0	True	True	True	False		
U[52]	Real	0.0	True	True	True	False		
U[53]	Real	0.0	True	True	True	False		
U[54]	Real	0.0	True	True	True	False		
U[55]	Real	0.0	True	True	True	False		
U[56]	Real	0.0	True	True	True	False		
U[57]	Real	0.0	True	True	True	False		
U[58]	Real	0.0	True	True	True	False		
U[59]	Real	0.0	True	True	True	False		
U[60]	Real	0.0	True	True	True	False		
U[61]	Real	0.0	True	True	True	False		
U[62]	Real	0.0	True	True	True	False		
U[63]	Real	0.0	True	True	True	False		
U[64]	Real	0.0	True	True	True	False		
U[65]	Real	0.0	True	True	True	False		
U[66]	Real	0.0	True	True	True	False		
U[67]	Real	0.0	True	True	True	False		
U[68]	Real	0.0	True	True	True	False		
U[69]	Real	0.0	True	True	True	False		
U[70]	Real	0.0	True	True	True	False		
U[71]	Real	0.0	True	True	True	False		
U[72]	Real	0.0	True	True	True	False		
U[73]	Real	0.0	True	True	True	False		
U[74]	Real	0.0	True	True	True	False		
U[75]	Real	0.0	True	True	True	False		
U[76]	Real	0.0	True	True	True	False		

Totally Integrated Automation Portal								
Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment	
U[77]	Real	0.0	True	True	True	False		
U[78]	Real	0.0	True	True	True	False		
U[79]	Real	0.0	True	True	True	False		
U[80]	Real	0.0	True	True	True	False		
U[81]	Real	0.0	True	True	True	False		
U[82]	Real	0.0	True	True	True	False		
U[83]	Real	0.0	True	True	True	False		
U[84]	Real	0.0	True	True	True	False		
U[85]	Real	0.0	True	True	True	False		
U[86]	Real	0.0	True	True	True	False		
U[87]	Real	0.0	True	True	True	False		
U[88]	Real	0.0	True	True	True	False		
U[89]	Real	0.0	True	True	True	False		
U[90]	Real	0.0	True	True	True	False		
U[91]	Real	0.0	True	True	True	False		
U[92]	Real	0.0	True	True	True	False		
U[93]	Real	0.0	True	True	True	False		
U[94]	Real	0.0	True	True	True	False		
U[95]	Real	0.0	True	True	True	False		
U[96]	Real	0.0	True	True	True	False		
U[97]	Real	0.0	True	True	True	False		
U[98]	Real	0.0	True	True	True	False		
U[99]	Real	0.0	True	True	True	False		
U[100]	Real	0.0	True	True	True	False		
U[101]	Real	0.0	True	True	True	False		
U[102]	Real	0.0	True	True	True	False		
U[103]	Real	0.0	True	True	True	False		
U[104]	Real	0.0	True	True	True	False		
U[105]	Real	0.0	True	True	True	False		
U[106]	Real	0.0	True	True	True	False		
U[107]	Real	0.0	True	True	True	False		
U[108]	Real	0.0	True	True	True	False		
U[109]	Real	0.0	True	True	True	False		
U[110]	Real	0.0	True	True	True	False		
U[111]	Real	0.0	True	True	True	False		
U[112]	Real	0.0	True	True	True	False		
U[113]	Real	0.0	True	True	True	False		
U[114]	Real	0.0	True	True	True	False		
U[115]	Real	0.0	True	True	True	False		
U[116]	Real	0.0	True	True	True	False		
U[117]	Real	0.0	True	True	True	False		
U[118]	Real	0.0	True	True	True	False		
U[119]	Real	0.0	True	True	True	False		
U[120]	Real	0.0	True	True	True	False		
U[121]	Real	0.0	True	True	True	False		
U[122]	Real	0.0	True	True	True	False		
U[123]	Real	0.0	True	True	True	False		
U[124]	Real	0.0	True	True	True	False		
U[125]	Real	0.0	True	True	True	False		
U[126]	Real	0.0	True	True	True	False		
U[127]	Real	0.0	True	True	True	False		
U[128]	Real	0.0	True	True	True	False		
U[129]	Real	0.0	True	True	True	False		
U[130]	Real	0.0	True	True	True	False		
U[131]	Real	0.0	True	True	True	False		
U[132]	Real	0.0	True	True	True	False		
U[133]	Real	0.0	True	True	True	False		
U[134]	Real	0.0	True	True	True	False		
U[135]	Real	0.0	True	True	True	False		
U[136]	Real	0.0	True	True	True	False		
U[137]	Real	0.0	True	True	True	False		
U[138]	Real	0.0	True	True	True	False		
U[139]	Real	0.0	True	True	True	False		
U[140]	Real	0.0	True	True	True	False		
U[141]	Real	0.0	True	True	True	False		
U[142]	Real	0.0	True	True	True	False		
U[143]	Real	0.0	True	True	True	False		
U[144]	Real	0.0	True	True	True	False		
U[145]	Real	0.0	True	True	True	False		
U[146]	Real	0.0	True	True	True	False		
U[147]	Real	0.0	True	True	True	False		
U[148]	Real	0.0	True	True	True	False		
U[149]	Real	0.0	True	True	True	False		
U[150]	Real	0.0	True	True	True	False		
U[151]	Real	0.0	True	True	True	False		
U[152]	Real	0.0	True	True	True	False		
U[153]	Real	0.0	True	True	True	False		
U[154]	Real	0.0	True	True	True	False		
U[155]	Real	0.0	True	True	True	False		
U[156]	Real	0.0	True	True	True	False		

Totally Integrated Automation Portal								
Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment	
U[157]	Real	0.0	True	True	True	False		
U[158]	Real	0.0	True	True	True	False		
U[159]	Real	0.0	True	True	True	False		
U[160]	Real	0.0	True	True	True	False		
U[161]	Real	0.0	True	True	True	False		
U[162]	Real	0.0	True	True	True	False		
U[163]	Real	0.0	True	True	True	False		
U[164]	Real	0.0	True	True	True	False		
U[165]	Real	0.0	True	True	True	False		
U[166]	Real	0.0	True	True	True	False		
U[167]	Real	0.0	True	True	True	False		
U[168]	Real	0.0	True	True	True	False		
U[169]	Real	0.0	True	True	True	False		
U[170]	Real	0.0	True	True	True	False		
U[171]	Real	0.0	True	True	True	False		
U[172]	Real	0.0	True	True	True	False		
U[173]	Real	0.0	True	True	True	False		
U[174]	Real	0.0	True	True	True	False		
U[175]	Real	0.0	True	True	True	False		
U[176]	Real	0.0	True	True	True	False		
U[177]	Real	0.0	True	True	True	False		
U[178]	Real	0.0	True	True	True	False		
U[179]	Real	0.0	True	True	True	False		
U[180]	Real	0.0	True	True	True	False		
U[181]	Real	0.0	True	True	True	False		
U[182]	Real	0.0	True	True	True	False		
U[183]	Real	0.0	True	True	True	False		
U[184]	Real	0.0	True	True	True	False		
U[185]	Real	0.0	True	True	True	False		
U[186]	Real	0.0	True	True	True	False		
U[187]	Real	0.0	True	True	True	False		
U[188]	Real	0.0	True	True	True	False		
U[189]	Real	0.0	True	True	True	False		
U[190]	Real	0.0	True	True	True	False		
U[191]	Real	0.0	True	True	True	False		
U[192]	Real	0.0	True	True	True	False		
U[193]	Real	0.0	True	True	True	False		
U[194]	Real	0.0	True	True	True	False		
U[195]	Real	0.0	True	True	True	False		
U[196]	Real	0.0	True	True	True	False		
U[197]	Real	0.0	True	True	True	False		
U[198]	Real	0.0	True	True	True	False		
U[199]	Real	0.0	True	True	True	False		
U[200]	Real	0.0	True	True	True	False		
U[201]	Real	0.0	True	True	True	False		
U[202]	Real	0.0	True	True	True	False		
U[203]	Real	0.0	True	True	True	False		
U[204]	Real	0.0	True	True	True	False		
U[205]	Real	0.0	True	True	True	False		
U[206]	Real	0.0	True	True	True	False		
U[207]	Real	0.0	True	True	True	False		
U[208]	Real	0.0	True	True	True	False		
U[209]	Real	0.0	True	True	True	False		
U[210]	Real	0.0	True	True	True	False		
U[211]	Real	0.0	True	True	True	False		
U[212]	Real	0.0	True	True	True	False		
U[213]	Real	0.0	True	True	True	False		
U[214]	Real	0.0	True	True	True	False		
U[215]	Real	0.0	True	True	True	False		
U[216]	Real	0.0	True	True	True	False		
U[217]	Real	0.0	True	True	True	False		
U[218]	Real	0.0	True	True	True	False		
U[219]	Real	0.0	True	True	True	False		
U[220]	Real	0.0	True	True	True	False		
U[221]	Real	0.0	True	True	True	False		
U[222]	Real	0.0	True	True	True	False		
U[223]	Real	0.0	True	True	True	False		
U[224]	Real	0.0	True	True	True	False		
U[225]	Real	0.0	True	True	True	False		
U[226]	Real	0.0	True	True	True	False		
U[227]	Real	0.0	True	True	True	False		
U[228]	Real	0.0	True	True	True	False		
U[229]	Real	0.0	True	True	True	False		
U[230]	Real	0.0	True	True	True	False		
U[231]	Real	0.0	True	True	True	False		
U[232]	Real	0.0	True	True	True	False		
U[233]	Real	0.0	True	True	True	False		
U[234]	Real	0.0	True	True	True	False		
U[235]	Real	0.0	True	True	True	False		
U[236]	Real	0.0	True	True	True	False		

Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
U[237]	Real	0.0	True	True	True	False	
U[238]	Real	0.0	True	True	True	False	
U[239]	Real	0.0	True	True	True	False	
U[240]	Real	0.0	True	True	True	False	
U[241]	Real	0.0	True	True	True	False	
U[242]	Real	0.0	True	True	True	False	
U[243]	Real	0.0	True	True	True	False	
U[244]	Real	0.0	True	True	True	False	
U[245]	Real	0.0	True	True	True	False	
U[246]	Real	0.0	True	True	True	False	
U[247]	Real	0.0	True	True	True	False	
U[248]	Real	0.0	True	True	True	False	
U[249]	Real	0.0	True	True	True	False	
U[250]	Real	0.0	True	True	True	False	
U[251]	Real	0.0	True	True	True	False	
U[252]	Real	0.0	True	True	True	False	
U[253]	Real	0.0	True	True	True	False	
U[254]	Real	0.0	True	True	True	False	
U[255]	Real	0.0	True	True	True	False	
U[256]	Real	0.0	True	True	True	False	
U[257]	Real	0.0	True	True	True	False	
U[258]	Real	0.0	True	True	True	False	
U[259]	Real	0.0	True	True	True	False	
U[260]	Real	0.0	True	True	True	False	
U[261]	Real	0.0	True	True	True	False	
U[262]	Real	0.0	True	True	True	False	
U[263]	Real	0.0	True	True	True	False	
U[264]	Real	0.0	True	True	True	False	
U[265]	Real	0.0	True	True	True	False	
U[266]	Real	0.0	True	True	True	False	
U[267]	Real	0.0	True	True	True	False	
U[268]	Real	0.0	True	True	True	False	
U[269]	Real	0.0	True	True	True	False	
U[270]	Real	0.0	True	True	True	False	
U[271]	Real	0.0	True	True	True	False	
U[272]	Real	0.0	True	True	True	False	
U[273]	Real	0.0	True	True	True	False	
U[274]	Real	0.0	True	True	True	False	
U[275]	Real	0.0	True	True	True	False	
U[276]	Real	0.0	True	True	True	False	
U[277]	Real	0.0	True	True	True	False	
U[278]	Real	0.0	True	True	True	False	
U[279]	Real	0.0	True	True	True	False	
U[280]	Real	0.0	True	True	True	False	
U[281]	Real	0.0	True	True	True	False	
U[282]	Real	0.0	True	True	True	False	
U[283]	Real	0.0	True	True	True	False	
U[284]	Real	0.0	True	True	True	False	
U[285]	Real	0.0	True	True	True	False	
U[286]	Real	0.0	True	True	True	False	
U[287]	Real	0.0	True	True	True	False	
U[288]	Real	0.0	True	True	True	False	
U[289]	Real	0.0	True	True	True	False	
U[290]	Real	0.0	True	True	True	False	
U[291]	Real	0.0	True	True	True	False	
U[292]	Real	0.0	True	True	True	False	
U[293]	Real	0.0	True	True	True	False	
U[294]	Real	0.0	True	True	True	False	
U[295]	Real	0.0	True	True	True	False	
U[296]	Real	0.0	True	True	True	False	
U[297]	Real	0.0	True	True	True	False	
U[298]	Real	0.0	True	True	True	False	
U[299]	Real	0.0	True	True	True	False	
U[300]	Real	0.0	True	True	True	False	

test_FO_s1200 / PSE [CPU 1511-1 PN] / Program blocks

PSE_factor_calc [FC1]

PSE_factor_calc Properties

General

Name	PSE_factor_calc	Number	1	Type	FC	Language	SCL
-------------	-----------------	---------------	---	-------------	----	-----------------	-----

Numbering	automatic
------------------	-----------

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

PSE_factor_calc

Name	Data type	Default value	Comment
▼ Input			
alfa	Real		factor
▼ Output			
▼ wsp_PSE	Array[1..200] of Real		
wsp_PSE[1]	Real		
wsp_PSE[2]	Real		
wsp_PSE[3]	Real		
wsp_PSE[4]	Real		
wsp_PSE[5]	Real		
wsp_PSE[6]	Real		
wsp_PSE[7]	Real		
wsp_PSE[8]	Real		
wsp_PSE[9]	Real		
wsp_PSE[10]	Real		
wsp_PSE[11]	Real		
wsp_PSE[12]	Real		
wsp_PSE[13]	Real		
wsp_PSE[14]	Real		
wsp_PSE[15]	Real		
wsp_PSE[16]	Real		
wsp_PSE[17]	Real		
wsp_PSE[18]	Real		
wsp_PSE[19]	Real		
wsp_PSE[20]	Real		
wsp_PSE[21]	Real		
wsp_PSE[22]	Real		
wsp_PSE[23]	Real		
wsp_PSE[24]	Real		
wsp_PSE[25]	Real		
wsp_PSE[26]	Real		
wsp_PSE[27]	Real		
wsp_PSE[28]	Real		
wsp_PSE[29]	Real		
wsp_PSE[30]	Real		
wsp_PSE[31]	Real		
wsp_PSE[32]	Real		
wsp_PSE[33]	Real		
wsp_PSE[34]	Real		
wsp_PSE[35]	Real		
wsp_PSE[36]	Real		
wsp_PSE[37]	Real		
wsp_PSE[38]	Real		
wsp_PSE[39]	Real		
wsp_PSE[40]	Real		
wsp_PSE[41]	Real		
wsp_PSE[42]	Real		
wsp_PSE[43]	Real		
wsp_PSE[44]	Real		
wsp_PSE[45]	Real		
wsp_PSE[46]	Real		
wsp_PSE[47]	Real		
wsp_PSE[48]	Real		
wsp_PSE[49]	Real		
wsp_PSE[50]	Real		
wsp_PSE[51]	Real		
wsp_PSE[52]	Real		
wsp_PSE[53]	Real		
wsp_PSE[54]	Real		
wsp_PSE[55]	Real		
wsp_PSE[56]	Real		
wsp_PSE[57]	Real		
wsp_PSE[58]	Real		
wsp_PSE[59]	Real		
wsp_PSE[60]	Real		
wsp_PSE[61]	Real		
wsp_PSE[62]	Real		
wsp_PSE[63]	Real		

Totally Integrated Automation Portal			
Name	Data type	Default value	Comment
wsp_PSE[64]	Real		
wsp_PSE[65]	Real		
wsp_PSE[66]	Real		
wsp_PSE[67]	Real		
wsp_PSE[68]	Real		
wsp_PSE[69]	Real		
wsp_PSE[70]	Real		
wsp_PSE[71]	Real		
wsp_PSE[72]	Real		
wsp_PSE[73]	Real		
wsp_PSE[74]	Real		
wsp_PSE[75]	Real		
wsp_PSE[76]	Real		
wsp_PSE[77]	Real		
wsp_PSE[78]	Real		
wsp_PSE[79]	Real		
wsp_PSE[80]	Real		
wsp_PSE[81]	Real		
wsp_PSE[82]	Real		
wsp_PSE[83]	Real		
wsp_PSE[84]	Real		
wsp_PSE[85]	Real		
wsp_PSE[86]	Real		
wsp_PSE[87]	Real		
wsp_PSE[88]	Real		
wsp_PSE[89]	Real		
wsp_PSE[90]	Real		
wsp_PSE[91]	Real		
wsp_PSE[92]	Real		
wsp_PSE[93]	Real		
wsp_PSE[94]	Real		
wsp_PSE[95]	Real		
wsp_PSE[96]	Real		
wsp_PSE[97]	Real		
wsp_PSE[98]	Real		
wsp_PSE[99]	Real		
wsp_PSE[100]	Real		
wsp_PSE[101]	Real		
wsp_PSE[102]	Real		
wsp_PSE[103]	Real		
wsp_PSE[104]	Real		
wsp_PSE[105]	Real		
wsp_PSE[106]	Real		
wsp_PSE[107]	Real		
wsp_PSE[108]	Real		
wsp_PSE[109]	Real		
wsp_PSE[110]	Real		
wsp_PSE[111]	Real		
wsp_PSE[112]	Real		
wsp_PSE[113]	Real		
wsp_PSE[114]	Real		
wsp_PSE[115]	Real		
wsp_PSE[116]	Real		
wsp_PSE[117]	Real		
wsp_PSE[118]	Real		
wsp_PSE[119]	Real		
wsp_PSE[120]	Real		
wsp_PSE[121]	Real		
wsp_PSE[122]	Real		
wsp_PSE[123]	Real		
wsp_PSE[124]	Real		
wsp_PSE[125]	Real		
wsp_PSE[126]	Real		
wsp_PSE[127]	Real		
wsp_PSE[128]	Real		
wsp_PSE[129]	Real		
wsp_PSE[130]	Real		
wsp_PSE[131]	Real		
wsp_PSE[132]	Real		
wsp_PSE[133]	Real		
wsp_PSE[134]	Real		
wsp_PSE[135]	Real		
wsp_PSE[136]	Real		
wsp_PSE[137]	Real		
wsp_PSE[138]	Real		
wsp_PSE[139]	Real		
wsp_PSE[140]	Real		
wsp_PSE[141]	Real		
wsp_PSE[142]	Real		
wsp_PSE[143]	Real		
wsp_PSE[144]	Real		

Name	Data type	Default value	Comment
wsp_PSE[145]	Real		
wsp_PSE[146]	Real		
wsp_PSE[147]	Real		
wsp_PSE[148]	Real		
wsp_PSE[149]	Real		
wsp_PSE[150]	Real		
wsp_PSE[151]	Real		
wsp_PSE[152]	Real		
wsp_PSE[153]	Real		
wsp_PSE[154]	Real		
wsp_PSE[155]	Real		
wsp_PSE[156]	Real		
wsp_PSE[157]	Real		
wsp_PSE[158]	Real		
wsp_PSE[159]	Real		
wsp_PSE[160]	Real		
wsp_PSE[161]	Real		
wsp_PSE[162]	Real		
wsp_PSE[163]	Real		
wsp_PSE[164]	Real		
wsp_PSE[165]	Real		
wsp_PSE[166]	Real		
wsp_PSE[167]	Real		
wsp_PSE[168]	Real		
wsp_PSE[169]	Real		
wsp_PSE[170]	Real		
wsp_PSE[171]	Real		
wsp_PSE[172]	Real		
wsp_PSE[173]	Real		
wsp_PSE[174]	Real		
wsp_PSE[175]	Real		
wsp_PSE[176]	Real		
wsp_PSE[177]	Real		
wsp_PSE[178]	Real		
wsp_PSE[179]	Real		
wsp_PSE[180]	Real		
wsp_PSE[181]	Real		
wsp_PSE[182]	Real		
wsp_PSE[183]	Real		
wsp_PSE[184]	Real		
wsp_PSE[185]	Real		
wsp_PSE[186]	Real		
wsp_PSE[187]	Real		
wsp_PSE[188]	Real		
wsp_PSE[189]	Real		
wsp_PSE[190]	Real		
wsp_PSE[191]	Real		
wsp_PSE[192]	Real		
wsp_PSE[193]	Real		
wsp_PSE[194]	Real		
wsp_PSE[195]	Real		
wsp_PSE[196]	Real		
wsp_PSE[197]	Real		
wsp_PSE[198]	Real		
wsp_PSE[199]	Real		
wsp_PSE[200]	Real		
InOut			
▼ Temp			
n	Int		
Constant			
▼ Return			
PSE_factor_calc	Void		

```

0001 #wsp_PSE[1] := 1;
0002
0003 FOR #n := 2 TO 200 DO
0004     #wsp_PSE[#n] := (1 - (1 + #alfa) / (#n - 1)) * #wsp_PSE[#n - 1];
0005 END_FOR;
0006
0007

```

Symbol	Address	Type	Comment
#alfa		Real	factor
#n		Int	
#wsp_PSE		Array	

test_FO_s1200 / PSE [CPU 1511-1 PN] / Program blocks

PSE [FB1]

PSE Properties

General

Name	PSE	Number	1	Type	FB	Language	SCL
Numbering	automatic						

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

PSE

Name	Data type	Default value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Input							
tmax	Int	50	Non-retain	True	True	False	
L	Int	10	Non-retain	True	True	False	
▼ Output							
▼ PSE	Array[1..300] of Real		Non-retain	True	True	False	
PSE[1]	Real	0.0	Non-retain	True	True	False	
PSE[2]	Real	0.0	Non-retain	True	True	False	
PSE[3]	Real	0.0	Non-retain	True	True	False	
PSE[4]	Real	0.0	Non-retain	True	True	False	
PSE[5]	Real	0.0	Non-retain	True	True	False	
PSE[6]	Real	0.0	Non-retain	True	True	False	
PSE[7]	Real	0.0	Non-retain	True	True	False	
PSE[8]	Real	0.0	Non-retain	True	True	False	
PSE[9]	Real	0.0	Non-retain	True	True	False	
PSE[10]	Real	0.0	Non-retain	True	True	False	
PSE[11]	Real	0.0	Non-retain	True	True	False	
PSE[12]	Real	0.0	Non-retain	True	True	False	
PSE[13]	Real	0.0	Non-retain	True	True	False	
PSE[14]	Real	0.0	Non-retain	True	True	False	
PSE[15]	Real	0.0	Non-retain	True	True	False	
PSE[16]	Real	0.0	Non-retain	True	True	False	
PSE[17]	Real	0.0	Non-retain	True	True	False	
PSE[18]	Real	0.0	Non-retain	True	True	False	
PSE[19]	Real	0.0	Non-retain	True	True	False	
PSE[20]	Real	0.0	Non-retain	True	True	False	
PSE[21]	Real	0.0	Non-retain	True	True	False	
PSE[22]	Real	0.0	Non-retain	True	True	False	
PSE[23]	Real	0.0	Non-retain	True	True	False	
PSE[24]	Real	0.0	Non-retain	True	True	False	
PSE[25]	Real	0.0	Non-retain	True	True	False	
PSE[26]	Real	0.0	Non-retain	True	True	False	
PSE[27]	Real	0.0	Non-retain	True	True	False	
PSE[28]	Real	0.0	Non-retain	True	True	False	
PSE[29]	Real	0.0	Non-retain	True	True	False	
PSE[30]	Real	0.0	Non-retain	True	True	False	
PSE[31]	Real	0.0	Non-retain	True	True	False	
PSE[32]	Real	0.0	Non-retain	True	True	False	
PSE[33]	Real	0.0	Non-retain	True	True	False	
PSE[34]	Real	0.0	Non-retain	True	True	False	
PSE[35]	Real	0.0	Non-retain	True	True	False	
PSE[36]	Real	0.0	Non-retain	True	True	False	
PSE[37]	Real	0.0	Non-retain	True	True	False	
PSE[38]	Real	0.0	Non-retain	True	True	False	
PSE[39]	Real	0.0	Non-retain	True	True	False	
PSE[40]	Real	0.0	Non-retain	True	True	False	
PSE[41]	Real	0.0	Non-retain	True	True	False	
PSE[42]	Real	0.0	Non-retain	True	True	False	
PSE[43]	Real	0.0	Non-retain	True	True	False	
PSE[44]	Real	0.0	Non-retain	True	True	False	
PSE[45]	Real	0.0	Non-retain	True	True	False	
PSE[46]	Real	0.0	Non-retain	True	True	False	
PSE[47]	Real	0.0	Non-retain	True	True	False	
PSE[48]	Real	0.0	Non-retain	True	True	False	
PSE[49]	Real	0.0	Non-retain	True	True	False	
PSE[50]	Real	0.0	Non-retain	True	True	False	
PSE[51]	Real	0.0	Non-retain	True	True	False	
PSE[52]	Real	0.0	Non-retain	True	True	False	
PSE[53]	Real	0.0	Non-retain	True	True	False	
PSE[54]	Real	0.0	Non-retain	True	True	False	
PSE[55]	Real	0.0	Non-retain	True	True	False	
PSE[56]	Real	0.0	Non-retain	True	True	False	
PSE[57]	Real	0.0	Non-retain	True	True	False	
PSE[58]	Real	0.0	Non-retain	True	True	False	
PSE[59]	Real	0.0	Non-retain	True	True	False	
PSE[60]	Real	0.0	Non-retain	True	True	False	

Totally Integrated Automation Portal							
Name	Data type	Default value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
InOut							
▼ Static							
▼ U	Array[1..300] of Real		Retain	True	True	False	
U[1]	Real	10.0	Retain	True	True	False	
U[2]	Real	10.0	Retain	True	True	False	
U[3]	Real	10.0	Retain	True	True	False	
U[4]	Real	10.0	Retain	True	True	False	
U[5]	Real	10.0	Retain	True	True	False	
U[6]	Real	10.0	Retain	True	True	False	
U[7]	Real	10.0	Retain	True	True	False	
U[8]	Real	10.0	Retain	True	True	False	
U[9]	Real	10.0	Retain	True	True	False	
U[10]	Real	10.0	Retain	True	True	False	
U[11]	Real	10.0	Retain	True	True	False	
U[12]	Real	10.0	Retain	True	True	False	
U[13]	Real	10.0	Retain	True	True	False	
U[14]	Real	10.0	Retain	True	True	False	
U[15]	Real	10.0	Retain	True	True	False	
U[16]	Real	10.0	Retain	True	True	False	
U[17]	Real	10.0	Retain	True	True	False	
U[18]	Real	10.0	Retain	True	True	False	
U[19]	Real	10.0	Retain	True	True	False	
U[20]	Real	10.0	Retain	True	True	False	
U[21]	Real	10.0	Retain	True	True	False	
U[22]	Real	10.0	Retain	True	True	False	
U[23]	Real	10.0	Retain	True	True	False	
U[24]	Real	10.0	Retain	True	True	False	
U[25]	Real	10.0	Retain	True	True	False	
U[26]	Real	10.0	Retain	True	True	False	
U[27]	Real	10.0	Retain	True	True	False	
U[28]	Real	10.0	Retain	True	True	False	
U[29]	Real	10.0	Retain	True	True	False	
U[30]	Real	10.0	Retain	True	True	False	
U[31]	Real	10.0	Retain	True	True	False	
U[32]	Real	10.0	Retain	True	True	False	
U[33]	Real	10.0	Retain	True	True	False	
U[34]	Real	10.0	Retain	True	True	False	
U[35]	Real	10.0	Retain	True	True	False	
U[36]	Real	10.0	Retain	True	True	False	
U[37]	Real	10.0	Retain	True	True	False	
U[38]	Real	10.0	Retain	True	True	False	
U[39]	Real	10.0	Retain	True	True	False	
U[40]	Real	10.0	Retain	True	True	False	
U[41]	Real	10.0	Retain	True	True	False	
U[42]	Real	10.0	Retain	True	True	False	
U[43]	Real	10.0	Retain	True	True	False	
U[44]	Real	10.0	Retain	True	True	False	
U[45]	Real	10.0	Retain	True	True	False	
U[46]	Real	10.0	Retain	True	True	False	
U[47]	Real	10.0	Retain	True	True	False	
U[48]	Real	10.0	Retain	True	True	False	
U[49]	Real	10.0	Retain	True	True	False	
U[50]	Real	10.0	Retain	True	True	False	
U[51]	Real	0.0	Retain	True	True	False	
U[52]	Real	0.0	Retain	True	True	False	
U[53]	Real	0.0	Retain	True	True	False	
U[54]	Real	0.0	Retain	True	True	False	
U[55]	Real	0.0	Retain	True	True	False	
U[56]	Real	0.0	Retain	True	True	False	
U[57]	Real	0.0	Retain	True	True	False	
U[58]	Real	0.0	Retain	True	True	False	
U[59]	Real	0.0	Retain	True	True	False	
U[60]	Real	0.0	Retain	True	True	False	
U[61]	Real	0.0	Retain	True	True	False	
U[62]	Real	0.0	Retain	True	True	False	
U[63]	Real	0.0	Retain	True	True	False	
U[64]	Real	0.0	Retain	True	True	False	
U[65]	Real	0.0	Retain	True	True	False	
U[66]	Real	0.0	Retain	True	True	False	
U[67]	Real	0.0	Retain	True	True	False	
U[68]	Real	0.0	Retain	True	True	False	
U[69]	Real	0.0	Retain	True	True	False	
U[70]	Real	0.0	Retain	True	True	False	
U[71]	Real	0.0	Retain	True	True	False	
U[72]	Real	0.0	Retain	True	True	False	
U[73]	Real	0.0	Retain	True	True	False	
U[74]	Real	0.0	Retain	True	True	False	
U[75]	Real	0.0	Retain	True	True	False	
U[76]	Real	0.0	Retain	True	True	False	

Totally Integrated Automation Portal							
Name	Data type	Default value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
U[77]	Real	0.0	Retain	True	True	False	
U[78]	Real	0.0	Retain	True	True	False	
U[79]	Real	0.0	Retain	True	True	False	
U[80]	Real	0.0	Retain	True	True	False	
U[81]	Real	0.0	Retain	True	True	False	
U[82]	Real	0.0	Retain	True	True	False	
U[83]	Real	0.0	Retain	True	True	False	
U[84]	Real	0.0	Retain	True	True	False	
U[85]	Real	0.0	Retain	True	True	False	
U[86]	Real	0.0	Retain	True	True	False	
U[87]	Real	0.0	Retain	True	True	False	
U[88]	Real	0.0	Retain	True	True	False	
U[89]	Real	0.0	Retain	True	True	False	
U[90]	Real	0.0	Retain	True	True	False	
U[91]	Real	0.0	Retain	True	True	False	
U[92]	Real	0.0	Retain	True	True	False	
U[93]	Real	0.0	Retain	True	True	False	
U[94]	Real	0.0	Retain	True	True	False	
U[95]	Real	0.0	Retain	True	True	False	
U[96]	Real	0.0	Retain	True	True	False	
U[97]	Real	0.0	Retain	True	True	False	
U[98]	Real	0.0	Retain	True	True	False	
U[99]	Real	0.0	Retain	True	True	False	
U[100]	Real	0.0	Retain	True	True	False	
U[101]	Real	0.0	Retain	True	True	False	
U[102]	Real	0.0	Retain	True	True	False	
U[103]	Real	0.0	Retain	True	True	False	
U[104]	Real	0.0	Retain	True	True	False	
U[105]	Real	0.0	Retain	True	True	False	
U[106]	Real	0.0	Retain	True	True	False	
U[107]	Real	0.0	Retain	True	True	False	
U[108]	Real	0.0	Retain	True	True	False	
U[109]	Real	0.0	Retain	True	True	False	
U[110]	Real	0.0	Retain	True	True	False	
U[111]	Real	0.0	Retain	True	True	False	
U[112]	Real	0.0	Retain	True	True	False	
U[113]	Real	0.0	Retain	True	True	False	
U[114]	Real	0.0	Retain	True	True	False	
U[115]	Real	0.0	Retain	True	True	False	
U[116]	Real	0.0	Retain	True	True	False	
U[117]	Real	0.0	Retain	True	True	False	
U[118]	Real	0.0	Retain	True	True	False	
U[119]	Real	0.0	Retain	True	True	False	
U[120]	Real	0.0	Retain	True	True	False	
U[121]	Real	0.0	Retain	True	True	False	
U[122]	Real	0.0	Retain	True	True	False	
U[123]	Real	0.0	Retain	True	True	False	
U[124]	Real	0.0	Retain	True	True	False	
U[125]	Real	0.0	Retain	True	True	False	
U[126]	Real	0.0	Retain	True	True	False	
U[127]	Real	0.0	Retain	True	True	False	
U[128]	Real	0.0	Retain	True	True	False	
U[129]	Real	0.0	Retain	True	True	False	
U[130]	Real	0.0	Retain	True	True	False	
U[131]	Real	0.0	Retain	True	True	False	
U[132]	Real	0.0	Retain	True	True	False	
U[133]	Real	0.0	Retain	True	True	False	
U[134]	Real	0.0	Retain	True	True	False	
U[135]	Real	0.0	Retain	True	True	False	
U[136]	Real	0.0	Retain	True	True	False	
U[137]	Real	0.0	Retain	True	True	False	
U[138]	Real	0.0	Retain	True	True	False	
U[139]	Real	0.0	Retain	True	True	False	
U[140]	Real	0.0	Retain	True	True	False	
U[141]	Real	0.0	Retain	True	True	False	
U[142]	Real	0.0	Retain	True	True	False	
U[143]	Real	0.0	Retain	True	True	False	
U[144]	Real	0.0	Retain	True	True	False	
U[145]	Real	0.0	Retain	True	True	False	
U[146]	Real	0.0	Retain	True	True	False	
U[147]	Real	0.0	Retain	True	True	False	
U[148]	Real	0.0	Retain	True	True	False	
U[149]	Real	0.0	Retain	True	True	False	
U[150]	Real	0.0	Retain	True	True	False	
U[151]	Real	0.0	Retain	True	True	False	
U[152]	Real	0.0	Retain	True	True	False	
U[153]	Real	0.0	Retain	True	True	False	
U[154]	Real	0.0	Retain	True	True	False	
U[155]	Real	0.0	Retain	True	True	False	
U[156]	Real	0.0	Retain	True	True	False	

Totally Integrated Automation Portal							
Name	Data type	Default value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
U[157]	Real	0.0	Retain	True	True	False	
U[158]	Real	0.0	Retain	True	True	False	
U[159]	Real	0.0	Retain	True	True	False	
U[160]	Real	0.0	Retain	True	True	False	
U[161]	Real	0.0	Retain	True	True	False	
U[162]	Real	0.0	Retain	True	True	False	
U[163]	Real	0.0	Retain	True	True	False	
U[164]	Real	0.0	Retain	True	True	False	
U[165]	Real	0.0	Retain	True	True	False	
U[166]	Real	0.0	Retain	True	True	False	
U[167]	Real	0.0	Retain	True	True	False	
U[168]	Real	0.0	Retain	True	True	False	
U[169]	Real	0.0	Retain	True	True	False	
U[170]	Real	0.0	Retain	True	True	False	
U[171]	Real	0.0	Retain	True	True	False	
U[172]	Real	0.0	Retain	True	True	False	
U[173]	Real	0.0	Retain	True	True	False	
U[174]	Real	0.0	Retain	True	True	False	
U[175]	Real	0.0	Retain	True	True	False	
U[176]	Real	0.0	Retain	True	True	False	
U[177]	Real	0.0	Retain	True	True	False	
U[178]	Real	0.0	Retain	True	True	False	
U[179]	Real	0.0	Retain	True	True	False	
U[180]	Real	0.0	Retain	True	True	False	
U[181]	Real	0.0	Retain	True	True	False	
U[182]	Real	0.0	Retain	True	True	False	
U[183]	Real	0.0	Retain	True	True	False	
U[184]	Real	0.0	Retain	True	True	False	
U[185]	Real	0.0	Retain	True	True	False	
U[186]	Real	0.0	Retain	True	True	False	
U[187]	Real	0.0	Retain	True	True	False	
U[188]	Real	0.0	Retain	True	True	False	
U[189]	Real	0.0	Retain	True	True	False	
U[190]	Real	0.0	Retain	True	True	False	
U[191]	Real	0.0	Retain	True	True	False	
U[192]	Real	0.0	Retain	True	True	False	
U[193]	Real	0.0	Retain	True	True	False	
U[194]	Real	0.0	Retain	True	True	False	
U[195]	Real	0.0	Retain	True	True	False	
U[196]	Real	0.0	Retain	True	True	False	
U[197]	Real	0.0	Retain	True	True	False	
U[198]	Real	0.0	Retain	True	True	False	
U[199]	Real	0.0	Retain	True	True	False	
U[200]	Real	0.0	Retain	True	True	False	
U[201]	Real	0.0	Retain	True	True	False	
U[202]	Real	0.0	Retain	True	True	False	
U[203]	Real	0.0	Retain	True	True	False	
U[204]	Real	0.0	Retain	True	True	False	
U[205]	Real	0.0	Retain	True	True	False	
U[206]	Real	0.0	Retain	True	True	False	
U[207]	Real	0.0	Retain	True	True	False	
U[208]	Real	0.0	Retain	True	True	False	
U[209]	Real	0.0	Retain	True	True	False	
U[210]	Real	0.0	Retain	True	True	False	
U[211]	Real	0.0	Retain	True	True	False	
U[212]	Real	0.0	Retain	True	True	False	
U[213]	Real	0.0	Retain	True	True	False	
U[214]	Real	0.0	Retain	True	True	False	
U[215]	Real	0.0	Retain	True	True	False	
U[216]	Real	0.0	Retain	True	True	False	
U[217]	Real	0.0	Retain	True	True	False	
U[218]	Real	0.0	Retain	True	True	False	
U[219]	Real	0.0	Retain	True	True	False	
U[220]	Real	0.0	Retain	True	True	False	
U[221]	Real	0.0	Retain	True	True	False	
U[222]	Real	0.0	Retain	True	True	False	
U[223]	Real	0.0	Retain	True	True	False	
U[224]	Real	0.0	Retain	True	True	False	
U[225]	Real	0.0	Retain	True	True	False	
U[226]	Real	0.0	Retain	True	True	False	
U[227]	Real	0.0	Retain	True	True	False	
U[228]	Real	0.0	Retain	True	True	False	
U[229]	Real	0.0	Retain	True	True	False	
U[230]	Real	0.0	Retain	True	True	False	
U[231]	Real	0.0	Retain	True	True	False	
U[232]	Real	0.0	Retain	True	True	False	
U[233]	Real	0.0	Retain	True	True	False	
U[234]	Real	0.0	Retain	True	True	False	
U[235]	Real	0.0	Retain	True	True	False	
U[236]	Real	0.0	Retain	True	True	False	

Totally Integrated Automation Portal							
Name	Data type	Default value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
U[237]	Real	0.0	Retain	True	True	False	
U[238]	Real	0.0	Retain	True	True	False	
U[239]	Real	0.0	Retain	True	True	False	
U[240]	Real	0.0	Retain	True	True	False	
U[241]	Real	0.0	Retain	True	True	False	
U[242]	Real	0.0	Retain	True	True	False	
U[243]	Real	0.0	Retain	True	True	False	
U[244]	Real	0.0	Retain	True	True	False	
U[245]	Real	0.0	Retain	True	True	False	
U[246]	Real	0.0	Retain	True	True	False	
U[247]	Real	0.0	Retain	True	True	False	
U[248]	Real	0.0	Retain	True	True	False	
U[249]	Real	0.0	Retain	True	True	False	
U[250]	Real	0.0	Retain	True	True	False	
U[251]	Real	0.0	Retain	True	True	False	
U[252]	Real	0.0	Retain	True	True	False	
U[253]	Real	0.0	Retain	True	True	False	
U[254]	Real	0.0	Retain	True	True	False	
U[255]	Real	0.0	Retain	True	True	False	
U[256]	Real	0.0	Retain	True	True	False	
U[257]	Real	0.0	Retain	True	True	False	
U[258]	Real	0.0	Retain	True	True	False	
U[259]	Real	0.0	Retain	True	True	False	
U[260]	Real	0.0	Retain	True	True	False	
U[261]	Real	0.0	Retain	True	True	False	
U[262]	Real	0.0	Retain	True	True	False	
U[263]	Real	0.0	Retain	True	True	False	
U[264]	Real	0.0	Retain	True	True	False	
U[265]	Real	0.0	Retain	True	True	False	
U[266]	Real	0.0	Retain	True	True	False	
U[267]	Real	0.0	Retain	True	True	False	
U[268]	Real	0.0	Retain	True	True	False	
U[269]	Real	0.0	Retain	True	True	False	
U[270]	Real	0.0	Retain	True	True	False	
U[271]	Real	0.0	Retain	True	True	False	
U[272]	Real	0.0	Retain	True	True	False	
U[273]	Real	0.0	Retain	True	True	False	
U[274]	Real	0.0	Retain	True	True	False	
U[275]	Real	0.0	Retain	True	True	False	
U[276]	Real	0.0	Retain	True	True	False	
U[277]	Real	0.0	Retain	True	True	False	
U[278]	Real	0.0	Retain	True	True	False	
U[279]	Real	0.0	Retain	True	True	False	
U[280]	Real	0.0	Retain	True	True	False	
U[281]	Real	0.0	Retain	True	True	False	
U[282]	Real	0.0	Retain	True	True	False	
U[283]	Real	0.0	Retain	True	True	False	
U[284]	Real	0.0	Retain	True	True	False	
U[285]	Real	0.0	Retain	True	True	False	
U[286]	Real	0.0	Retain	True	True	False	
U[287]	Real	0.0	Retain	True	True	False	
U[288]	Real	0.0	Retain	True	True	False	
U[289]	Real	0.0	Retain	True	True	False	
U[290]	Real	0.0	Retain	True	True	False	
U[291]	Real	0.0	Retain	True	True	False	
U[292]	Real	0.0	Retain	True	True	False	
U[293]	Real	0.0	Retain	True	True	False	
U[294]	Real	0.0	Retain	True	True	False	
U[295]	Real	0.0	Retain	True	True	False	
U[296]	Real	0.0	Retain	True	True	False	
U[297]	Real	0.0	Retain	True	True	False	
U[298]	Real	0.0	Retain	True	True	False	
U[299]	Real	0.0	Retain	True	True	False	
U[300]	Real	0.0	Retain	True	True	False	
▼ Temp							
n	Int						
n_internal	Int						
n_sum	Int						
▼ temp	Array[1..300] of Real						
temp[1]	Real						
temp[2]	Real						
temp[3]	Real						
temp[4]	Real						
temp[5]	Real						
temp[6]	Real						
temp[7]	Real						
temp[8]	Real						
temp[9]	Real						
temp[10]	Real						

Totally Integrated Automation Portal								
Name	Data type	Default value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment	
temp[11]	Real							
temp[12]	Real							
temp[13]	Real							
temp[14]	Real							
temp[15]	Real							
temp[16]	Real							
temp[17]	Real							
temp[18]	Real							
temp[19]	Real							
temp[20]	Real							
temp[21]	Real							
temp[22]	Real							
temp[23]	Real							
temp[24]	Real							
temp[25]	Real							
temp[26]	Real							
temp[27]	Real							
temp[28]	Real							
temp[29]	Real							
temp[30]	Real							
temp[31]	Real							
temp[32]	Real							
temp[33]	Real							
temp[34]	Real							
temp[35]	Real							
temp[36]	Real							
temp[37]	Real							
temp[38]	Real							
temp[39]	Real							
temp[40]	Real							
temp[41]	Real							
temp[42]	Real							
temp[43]	Real							
temp[44]	Real							
temp[45]	Real							
temp[46]	Real							
temp[47]	Real							
temp[48]	Real							
temp[49]	Real							
temp[50]	Real							
temp[51]	Real							
temp[52]	Real							
temp[53]	Real							
temp[54]	Real							
temp[55]	Real							
temp[56]	Real							
temp[57]	Real							
temp[58]	Real							
temp[59]	Real							
temp[60]	Real							
temp[61]	Real							
temp[62]	Real							
temp[63]	Real							
temp[64]	Real							
temp[65]	Real							
temp[66]	Real							
temp[67]	Real							
temp[68]	Real							
temp[69]	Real							
temp[70]	Real							
temp[71]	Real							
temp[72]	Real							
temp[73]	Real							
temp[74]	Real							
temp[75]	Real							
temp[76]	Real							
temp[77]	Real							
temp[78]	Real							
temp[79]	Real							
temp[80]	Real							
temp[81]	Real							
temp[82]	Real							
temp[83]	Real							
temp[84]	Real							
temp[85]	Real							
temp[86]	Real							
temp[87]	Real							
temp[88]	Real							
temp[89]	Real							
temp[90]	Real							

Totally Integrated Automation Portal								
Name	Data type	Default value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment	
temp[91]	Real							
temp[92]	Real							
temp[93]	Real							
temp[94]	Real							
temp[95]	Real							
temp[96]	Real							
temp[97]	Real							
temp[98]	Real							
temp[99]	Real							
temp[100]	Real							
temp[101]	Real							
temp[102]	Real							
temp[103]	Real							
temp[104]	Real							
temp[105]	Real							
temp[106]	Real							
temp[107]	Real							
temp[108]	Real							
temp[109]	Real							
temp[110]	Real							
temp[111]	Real							
temp[112]	Real							
temp[113]	Real							
temp[114]	Real							
temp[115]	Real							
temp[116]	Real							
temp[117]	Real							
temp[118]	Real							
temp[119]	Real							
temp[120]	Real							
temp[121]	Real							
temp[122]	Real							
temp[123]	Real							
temp[124]	Real							
temp[125]	Real							
temp[126]	Real							
temp[127]	Real							
temp[128]	Real							
temp[129]	Real							
temp[130]	Real							
temp[131]	Real							
temp[132]	Real							
temp[133]	Real							
temp[134]	Real							
temp[135]	Real							
temp[136]	Real							
temp[137]	Real							
temp[138]	Real							
temp[139]	Real							
temp[140]	Real							
temp[141]	Real							
temp[142]	Real							
temp[143]	Real							
temp[144]	Real							
temp[145]	Real							
temp[146]	Real							
temp[147]	Real							
temp[148]	Real							
temp[149]	Real							
temp[150]	Real							
temp[151]	Real							
temp[152]	Real							
temp[153]	Real							
temp[154]	Real							
temp[155]	Real							
temp[156]	Real							
temp[157]	Real							
temp[158]	Real							
temp[159]	Real							
temp[160]	Real							
temp[161]	Real							
temp[162]	Real							
temp[163]	Real							
temp[164]	Real							
temp[165]	Real							
temp[166]	Real							
temp[167]	Real							
temp[168]	Real							
temp[169]	Real							
temp[170]	Real							

Totally Integrated Automation Portal								
Name	Data type	Default value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment	
temp[171]	Real							
temp[172]	Real							
temp[173]	Real							
temp[174]	Real							
temp[175]	Real							
temp[176]	Real							
temp[177]	Real							
temp[178]	Real							
temp[179]	Real							
temp[180]	Real							
temp[181]	Real							
temp[182]	Real							
temp[183]	Real							
temp[184]	Real							
temp[185]	Real							
temp[186]	Real							
temp[187]	Real							
temp[188]	Real							
temp[189]	Real							
temp[190]	Real							
temp[191]	Real							
temp[192]	Real							
temp[193]	Real							
temp[194]	Real							
temp[195]	Real							
temp[196]	Real							
temp[197]	Real							
temp[198]	Real							
temp[199]	Real							
temp[200]	Real							
temp[201]	Real							
temp[202]	Real							
temp[203]	Real							
temp[204]	Real							
temp[205]	Real							
temp[206]	Real							
temp[207]	Real							
temp[208]	Real							
temp[209]	Real							
temp[210]	Real							
temp[211]	Real							
temp[212]	Real							
temp[213]	Real							
temp[214]	Real							
temp[215]	Real							
temp[216]	Real							
temp[217]	Real							
temp[218]	Real							
temp[219]	Real							
temp[220]	Real							
temp[221]	Real							
temp[222]	Real							
temp[223]	Real							
temp[224]	Real							
temp[225]	Real							
temp[226]	Real							
temp[227]	Real							
temp[228]	Real							
temp[229]	Real							
temp[230]	Real							
temp[231]	Real							
temp[232]	Real							
temp[233]	Real							
temp[234]	Real							
temp[235]	Real							
temp[236]	Real							
temp[237]	Real							
temp[238]	Real							
temp[239]	Real							
temp[240]	Real							
temp[241]	Real							
temp[242]	Real							
temp[243]	Real							
temp[244]	Real							
temp[245]	Real							
temp[246]	Real							
temp[247]	Real							
temp[248]	Real							
temp[249]	Real							
temp[250]	Real							

Name	Data type	Default value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
temp[251]	Real						
temp[252]	Real						
temp[253]	Real						
temp[254]	Real						
temp[255]	Real						
temp[256]	Real						
temp[257]	Real						
temp[258]	Real						
temp[259]	Real						
temp[260]	Real						
temp[261]	Real						
temp[262]	Real						
temp[263]	Real						
temp[264]	Real						
temp[265]	Real						
temp[266]	Real						
temp[267]	Real						
temp[268]	Real						
temp[269]	Real						
temp[270]	Real						
temp[271]	Real						
temp[272]	Real						
temp[273]	Real						
temp[274]	Real						
temp[275]	Real						
temp[276]	Real						
temp[277]	Real						
temp[278]	Real						
temp[279]	Real						
temp[280]	Real						
temp[281]	Real						
temp[282]	Real						
temp[283]	Real						
temp[284]	Real						
temp[285]	Real						
temp[286]	Real						
temp[287]	Real						
temp[288]	Real						
temp[289]	Real						
temp[290]	Real						
temp[291]	Real						
temp[292]	Real						
temp[293]	Real						
temp[294]	Real						
temp[295]	Real						
temp[296]	Real						
temp[297]	Real						
temp[298]	Real						
temp[299]	Real						
temp[300]	Real						
Constant							

```

0001
0002 FOR #n:=1 TO #L DO
0003   FOR #n_internal := 1 TO #n DO
0004     #temp[#n_internal]:=#U[#n-#n_internal+1]*"PSE_wsp".PSE_factor[#n_internal];
0005   END_FOR;
0006   #PSE[#n] := 0;
0007   FOR #n_sum := 1 TO #n DO
0008     #PSE[#n]:=#PSE[#n]+#temp[#n_sum];
0009   END_FOR;
0010
0011 END_FOR;
0012
0013 FOR #n := #L+1 TO #tmax DO
0014   FOR #n_internal := 1 TO #L DO
0015     #temp[#n_internal] := #U[#n - #n_internal + 1] * "PSE_wsp".PSE_factor[#n_internal];
0016   END_FOR;
0017   #PSE[#n] := 0;
0018   FOR #n_sum := 1 TO #L DO
0019     #PSE[#n] := #PSE[#n] + #temp[#n_sum];
0020   END_FOR;
0021 END_FOR;

```

Symbol	Address	Type	Comment
"PSE_wsp".PSE_factor		Array	
#L		Int	
#n		Int	
#n_internal		Int	
#n_sum		Int	
#PSE		Array	
#temp		Array	
#tmax		Int	

test_FO_s1200 / PSE [CPU 1511-1 PN] / Program blocks

PSE_wsp [DB1]

PSE_wsp Properties

General

Name	PSE_wsp	Number	1	Type	DB	Language	DB
------	---------	--------	---	------	----	----------	----

Numbering	automatic
-----------	-----------

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

PSE_wsp

Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Static							
alfa	Real	-0.25	True	True	True	False	
▼ PSE_factor	Array[1..200] of Real		True	True	True	False	
PSE_factor[1]	Real	16.0	True	True	True	False	
PSE_factor[2]	Real	-32.0	True	True	True	False	
PSE_factor[3]	Real	-12.0	True	True	True	False	
PSE_factor[4]	Real	6.0	True	True	True	False	
PSE_factor[5]	Real	1.0	True	True	True	False	
PSE_factor[6]	Real	0.0	True	True	True	False	
PSE_factor[7]	Real	0.0	True	True	True	False	
PSE_factor[8]	Real	0.0	True	True	True	False	
PSE_factor[9]	Real	0.0	True	True	True	False	
PSE_factor[10]	Real	0.0	True	True	True	False	
PSE_factor[11]	Real	0.0	True	True	True	False	
PSE_factor[12]	Real	0.0	True	True	True	False	
PSE_factor[13]	Real	0.0	True	True	True	False	
PSE_factor[14]	Real	0.0	True	True	True	False	
PSE_factor[15]	Real	0.0	True	True	True	False	
PSE_factor[16]	Real	0.0	True	True	True	False	
PSE_factor[17]	Real	0.0	True	True	True	False	
PSE_factor[18]	Real	0.0	True	True	True	False	
PSE_factor[19]	Real	0.0	True	True	True	False	
PSE_factor[20]	Real	0.0	True	True	True	False	
PSE_factor[21]	Real	0.0	True	True	True	False	
PSE_factor[22]	Real	0.0	True	True	True	False	
PSE_factor[23]	Real	0.0	True	True	True	False	
PSE_factor[24]	Real	0.0	True	True	True	False	
PSE_factor[25]	Real	0.0	True	True	True	False	
PSE_factor[26]	Real	0.0	True	True	True	False	
PSE_factor[27]	Real	0.0	True	True	True	False	
PSE_factor[28]	Real	0.0	True	True	True	False	
PSE_factor[29]	Real	0.0	True	True	True	False	
PSE_factor[30]	Real	0.0	True	True	True	False	
PSE_factor[31]	Real	0.0	True	True	True	False	
PSE_factor[32]	Real	0.0	True	True	True	False	
PSE_factor[33]	Real	0.0	True	True	True	False	
PSE_factor[34]	Real	0.0	True	True	True	False	
PSE_factor[35]	Real	0.0	True	True	True	False	
PSE_factor[36]	Real	0.0	True	True	True	False	
PSE_factor[37]	Real	0.0	True	True	True	False	
PSE_factor[38]	Real	0.0	True	True	True	False	
PSE_factor[39]	Real	0.0	True	True	True	False	
PSE_factor[40]	Real	0.0	True	True	True	False	
PSE_factor[41]	Real	0.0	True	True	True	False	
PSE_factor[42]	Real	0.0	True	True	True	False	
PSE_factor[43]	Real	0.0	True	True	True	False	
PSE_factor[44]	Real	0.0	True	True	True	False	
PSE_factor[45]	Real	0.0	True	True	True	False	
PSE_factor[46]	Real	0.0	True	True	True	False	
PSE_factor[47]	Real	0.0	True	True	True	False	
PSE_factor[48]	Real	0.0	True	True	True	False	
PSE_factor[49]	Real	0.0	True	True	True	False	
PSE_factor[50]	Real	0.0	True	True	True	False	
PSE_factor[51]	Real	0.0	True	True	True	False	
PSE_factor[52]	Real	0.0	True	True	True	False	
PSE_factor[53]	Real	0.0	True	True	True	False	
PSE_factor[54]	Real	0.0	True	True	True	False	
PSE_factor[55]	Real	0.0	True	True	True	False	
PSE_factor[56]	Real	0.0	True	True	True	False	
PSE_factor[57]	Real	0.0	True	True	True	False	
PSE_factor[58]	Real	0.0	True	True	True	False	
PSE_factor[59]	Real	0.0	True	True	True	False	
PSE_factor[60]	Real	0.0	True	True	True	False	
PSE_factor[61]	Real	0.0	True	True	True	False	
PSE_factor[62]	Real	0.0	True	True	True	False	
PSE_factor[63]	Real	0.0	True	True	True	False	

Totally Integrated Automation Portal							
Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
PSE_factor[144]	Real	0.0	True	True	True	False	
PSE_factor[145]	Real	0.0	True	True	True	False	
PSE_factor[146]	Real	0.0	True	True	True	False	
PSE_factor[147]	Real	0.0	True	True	True	False	
PSE_factor[148]	Real	0.0	True	True	True	False	
PSE_factor[149]	Real	0.0	True	True	True	False	
PSE_factor[150]	Real	0.0	True	True	True	False	
PSE_factor[151]	Real	0.0	True	True	True	False	
PSE_factor[152]	Real	0.0	True	True	True	False	
PSE_factor[153]	Real	0.0	True	True	True	False	
PSE_factor[154]	Real	0.0	True	True	True	False	
PSE_factor[155]	Real	0.0	True	True	True	False	
PSE_factor[156]	Real	0.0	True	True	True	False	
PSE_factor[157]	Real	0.0	True	True	True	False	
PSE_factor[158]	Real	0.0	True	True	True	False	
PSE_factor[159]	Real	0.0	True	True	True	False	
PSE_factor[160]	Real	0.0	True	True	True	False	
PSE_factor[161]	Real	0.0	True	True	True	False	
PSE_factor[162]	Real	0.0	True	True	True	False	
PSE_factor[163]	Real	0.0	True	True	True	False	
PSE_factor[164]	Real	0.0	True	True	True	False	
PSE_factor[165]	Real	0.0	True	True	True	False	
PSE_factor[166]	Real	0.0	True	True	True	False	
PSE_factor[167]	Real	0.0	True	True	True	False	
PSE_factor[168]	Real	0.0	True	True	True	False	
PSE_factor[169]	Real	0.0	True	True	True	False	
PSE_factor[170]	Real	0.0	True	True	True	False	
PSE_factor[171]	Real	0.0	True	True	True	False	
PSE_factor[172]	Real	0.0	True	True	True	False	
PSE_factor[173]	Real	0.0	True	True	True	False	
PSE_factor[174]	Real	0.0	True	True	True	False	
PSE_factor[175]	Real	0.0	True	True	True	False	
PSE_factor[176]	Real	0.0	True	True	True	False	
PSE_factor[177]	Real	0.0	True	True	True	False	
PSE_factor[178]	Real	0.0	True	True	True	False	
PSE_factor[179]	Real	0.0	True	True	True	False	
PSE_factor[180]	Real	0.0	True	True	True	False	
PSE_factor[181]	Real	0.0	True	True	True	False	
PSE_factor[182]	Real	0.0	True	True	True	False	
PSE_factor[183]	Real	0.0	True	True	True	False	
PSE_factor[184]	Real	0.0	True	True	True	False	
PSE_factor[185]	Real	0.0	True	True	True	False	
PSE_factor[186]	Real	0.0	True	True	True	False	
PSE_factor[187]	Real	0.0	True	True	True	False	
PSE_factor[188]	Real	0.0	True	True	True	False	
PSE_factor[189]	Real	0.0	True	True	True	False	
PSE_factor[190]	Real	0.0	True	True	True	False	
PSE_factor[191]	Real	0.0	True	True	True	False	
PSE_factor[192]	Real	0.0	True	True	True	False	
PSE_factor[193]	Real	0.0	True	True	True	False	
PSE_factor[194]	Real	0.0	True	True	True	False	
PSE_factor[195]	Real	0.0	True	True	True	False	
PSE_factor[196]	Real	0.0	True	True	True	False	
PSE_factor[197]	Real	0.0	True	True	True	False	
PSE_factor[198]	Real	0.0	True	True	True	False	
PSE_factor[199]	Real	0.0	True	True	True	False	
PSE_factor[200]	Real	0.0	True	True	True	False	

Dodatek C: Algorytm kukułki

W obliczenia wykonywanych w głównej części pracy, w których konieczna była optymalizacja wartości paramerów poszczególnych modeli dla rozważanego procesu cieplnego, wykorzystywano deterministyczną metodę wyboru elementu optymalnego przy użyciu algorytmu Nelder-Meada z użyciem funkcji $fminsearch$ zaimplementowaną w środowisku Matlab.

Dodatek ten zawiera wyniki badań otrzymane z wykorzystaniem jednej z niedeterministycznych metod optymalizacji - algorytm kukułki (Cuckoo Search CS), który należy do grupy metod heurystycznych inspirowanych naturą ([90],[91], [92]). W ostatnich latach możemy zaobserwować dynamiczny rozwój algorytmów stadnych, które odzwierciedlają naturalne procesy, które zachodzą w systemach naturalnych. Znane są prace, w których wykorzystywano do modelowania procesów cieplnych algorytmy optymalizacji rojem cząstek (Particle Swarm Optimization PSO) [93] czy też algorytmem szarych wilków (Grey Wolf Optimizer GWO) lub algorytmem optymalizacji rojem karaluchów (Cockroach Swarm Optimization CSO) [94].

Parametry proponowanych w rozprawie dyskretenych modeli transmitancji niecałkowitego rzędu z opóźnieniem opisanych równaniami (3.65) i (3.65) oraz porównywanych znanych modeli zastępczych Kupfmuellera całkowitego rzędu (2.14) obliczone przy użyciu optymalizacji funkcji kosztu (3.69) z wykorzystaniem algorytmu kukułki są podane w tabelach 15 oraz 16.

Tabela 15. Optymalne parametry i funkcja kosztów MSE (3.69) dla modelu (3.65) oraz dla modelu całkowitego rzędu (2.14) otrzymane z użyciem algorytmu kukułki

Nr czujnika	α ,	T_α ,	N_1 ,	MSE	T_1 ,	N_1 ,	MSE
1	1.2183	38.0751	8	0.0944	29.0423	10	0.1133
2	1.5431	99.1101	16	0.0093	43.6374	23	0.0301
3	0.0168	1.8346	28	0.0033	68.6097	40	0.0060

Otrzymane wyniki (tabele 15 oraz 16) są porównywalne z wcześniej otrzymanymi wartościami w sensie wyboru elementu optymalnego (tabele 3.13 oraz 3.14), co potwierdza iż nowe,

Tabela 16. Optymalne parametry i funkcja kosztów MSE (3.69) dla modelu (3.65) oraz dla modelu całkowitego rzędu (2.14) otrzymane z użyciem algorytmu kukułki

Nr czujnika	$\alpha,$	$\beta,$	T_α	T_β	$N_2,$	MSE		T_{21}	T_{22}	N_2	MSE
1	0.0417	0.0060	0.0148	6.96558	3	0.0417		3.3175	28.5624	7	0.0009
2	0.0071	0.0161	6.4303	0.6829	8	10.5969		40.1456	40.1456	15	0.0089
3	0.0179	0.0088	2.0432	1.7355	10	0.0021		61.4630	17.4610	27	0.0033

zaproponowane w rozprawie modele niecałkowitego rzędu są w stanie dokładniej opisać rozkład temperatury w rozważanym obiekcie doświadczalnym, niż znane wcześniej modele transmitancji całkowitego rzędu z opóźnieniem.

Bibliografia

- [1] B. Baeumer, S. Kurita i M. Meerschaert. „Inhomogeneous Fractional Diffusion Equations”. W: *Fractional Calculus and Applied Analysis* 8.4 (2005), s. 371–386.
- [2] A. Kochubei. *Fractional-Parabolic Systems, preprint, arXiv:1009.4996 [math.AP]*. 2011.
- [3] R. Almeida i D. F. M. Torres. „Necessary and sufficient conditions for the fractional calculus of variations with Caputo derivatives”. W: *Communications in Nonlinear Science and Numerical Simulation* 16.3 (2011), s. 1490–1500.
- [4] I. Podlubny. *Fractional Differential Equations*. T. 198. San Diego: Academic Press, 1999.
- [5] A. Dzielinski, D. Sierociuk i G. Sarwas. „Some applications of fractional order calculus”. W: *Bulletin of the Polish Academy of Sciences, Technical Sciences* 58.4 (2010), s. 583–592.
- [6] R. Caponetto i in. „Fractional Order Systems: Modeling and Control Applications”. W: *Series on Nonlinear Science*. Red. Leon O. Chua. Berkeley: University of California, 2010, s. 1–178.
- [7] S. Das. *Functional Fractional Calculus for System Identification and Control*. Berlin: Springer, 2010.
- [8] A. Obraczka. „Control of heat processes with the use of non-integer models”. Prac. dokt. Krakow, Poland: AGH University, 2014.
- [9] D. Sierociuk i in. „Diffusion process modeling by using fractional-order models”. W: *Applied Mathematics and Computation* 257.1 (2015), s. 2–11.
- [10] K. Burnecki. „Identification, validation and prediction of fractional dynamic systems”. W: *Oficyna Wydawnicza Politechniki Wrocławskiej* (2012), s. 1–125.
- [11] T. Kaczorek. „Positive stable realizations of fractional continuous-time linear systems”. W: *International Journal of Applied Mathematics and Computer Science* 21.4 (2011), s. 697–702.

- [12] T. Kaczorek. *Selected Problems of Fractional Systems Theory*. Berlin: Springer, 2011.
- [13] W. Mitkowski i A. Obraczka. „Simple identification of fractional differential equation”. W: *Solid State Phenomena* 1.180 (2012), s. 331–338.
- [14] A. Obraczka i W. Mitkowski. „The comparison of parameter identification methods for fractional partial differential equation”. W: *Solid State Phenomena* 210 (2014), s. 265–270.
- [15] E. Popescu. „On the fractional Cauchy problem associated with a Feller semigroup”. W: *Mathematical Reports* 12.2 (2010), s. 181–188.
- [16] W. Mitkowski. „Approximation of fractional diffusion-wave equation”. W: *Acta Mechanica et Automatica* 5.2 (2011), s. 65–68.
- [17] M. Dlugosz i P. Skruch. „The application of fractional-order models for thermal process modelling inside buildings”. W: *Journal of Building Physics* 1.1 (2015), s. 1–13.
- [18] S. Domek. „Switched Fractional State-Space Predictive Control Methods for Non-Linear Fractional Systems”. W: *Advances in Non-Integer Order Calculus and Its Applications* (2020).
- [19] K. Oprzedkiewicz, W. Mitkowski i E. Gawin. „Application of fractional order transfer functions to modeling of high - order systems”. W: *Advances in modelling and control of non - integer order systems: MMAR 2015: 20th International Conference on Methods and Models in Automation and Robotics, Międzyzdroje, Poland* (2015), s. 1169–1174.
- [20] K. Oprzedkiewicz, E. Gawin i W. Mitkowski. „Modeling heat distribution with the use of a non-integer order, state space model”. W: *International Journal of Applied Mathematics and Computer Science* 26.4 (2016), s. 749–756.
- [21] K. Oprzedkiewicz i E. Gawin. „A non integer order, state space model for one dimensional heat transfer process”. W: *Archives of Control Sciences* 26.2 (2016), s. 261–275.
- [22] K. Oprzedkiewicz, W. Mitkowski i E. Gawin. „An estimation of accuracy of Oustaloup approximation”. W: *Challenges in Automation, Robotics and Measurement Techniques, Advances in Intelligent Systems and Computing* (2016), s. 299–305.
- [23] K. Oprzedkiewicz, W. Mitkowski i E. Gawin. „Parameter identification for non integer order, state space models of heat plant”. W: *Advances in modelling and control of non - integer order systems: MMAR 2016: 21th International Conference on Methods and Models in Automation and Robotics, Międzyzdroje, Poland* (2016), s. 184–188.

- [24] K. Oprzedkiewicz i E. Gawin. „Modeling of heat transfer proces with the use of non integer order, discrete, transfer function models”. W: *Trends in Advanced Intelligent Control, Optimization and Automation 577* (2017), s. 425–433.
- [25] K. Oprzedkiewicz, E. Gawin i T. Gawin. „PLC implementations of an elementary fractional order operator”. W: *Conference: AUTSYM 2017, Wismar* (2017).
- [26] K. Oprzedkiewicz, W. Mitkowski i E. Gawin. „Accuracy analysis for new, state space, non integer order model of heat transfer process using Caputo-Fabrizio operator”. W: *International Conference on Fractional Signal Systems* (2017), s. 335–354.
- [27] K. Oprzedkiewicz, W. Mitkowski i E. Gawin. „Accuracy analysis for new, state space, non integer order model of heat transfer process using Caputo-Fabrizio operator”. W: *International Conference on Fractional Signal Systems* (2017).
- [28] K. Oprzedkiewicz i in. „A new algorithm for a CFE-approximated solution of a discrete-time noninteger-order state equation”. W: *Bulletin of the Polish Academy of Sciences Technical Sciences 65.4* (2017), s. 429–437.
- [29] K. Oprzedkiewicz i E. Gawin. „The practical stability of the discrete, fractional order, state space model of the heat transfer process”. W: *Archives of Control Sciences 28.3* (2018), s. 463–482.
- [30] K. Oprzedkiewicz, W. Mitkowski i E. Gawin. „The PLC implementation of fractional-order operator using CFE approximation”. W: *Automation 2017: Innovations in Automation, Robotics and Measurement Techniques 550* (2018), s. 22–33.
- [31] K. Oprzedkiewicz, W. Mitkowski i E. Gawin. „An Accuracy Estimation for a Non Integer Order, Discrete, State Space Model of Heat Transfer Process”. W: *Automation 2017: Innovations in Automation, Robotics and Measurement Techniques 550* (2018), s. 86–98.
- [32] K. Oprzedkiewicz i in. „The Caputo vs. Caputo-Fabrizio operators in modeling of heat transfer process”. W: *Bulletin of the Polish Academy of Sciences. Technical Sciences 66.4* (2018), s. 501–507.
- [33] K. Oprzedkiewicz, E. Gawin i T. Gawin. „Real-Time PLC Implementations of Fractional Order Operator”. W: *Automation 2018 : advances in automation, robotics and measurement techniques 743* (2018), s. 36–51.
- [34] K. Oprzedkiewicz, E. Gawin i W. Mitkowski. „A PLC implementation of PSE approximant for Fractional Order operator”. W: *Non-integer order calculus and its applications : 9th international conference on Non-integer order calculus and its applications 496* (2019), s. 102–112.

- [35] T. Kaczorek i K. Rogowski. *Fractional Linear Systems and Electrical Circuits*. Białystok: Białystok University of Technology, 2014.
- [36] P. Ostalczyk. „Equivalent descriptions of a discrete-time fractional-order linear system and its stability domains”. W: *International Journal of Applied Mathematics and Computer Science* 22.3 (2012), s. 533–538.
- [37] T. Kaczorek. „Reduced-order fractional descriptor observers for a class of fractional descriptor continuous-time nonlinear systems”. W: *International Journal of Applied Mathematics and Computer Science* 26.2 (2016), s. 277–283.
- [38] M. Caputo i M. Fabrizio. „A new definition of fractional derivative without singular kernel”. W: *Progress in Fractional Differentiation and Applications* 1.2 (2015), s. 1–13.
- [39] A. Atangana i D. Baleanu. „New fractional derivatives with non-local and non-singular kernel: theory and application to heat transfer”. W: *Thermal Sciences* 20.2 (2016), 763–769.
- [40] T. Kaczorek i K. Borawski. „Fractional descriptor continuous-time linear systems described by the Caputo-Fabrizio derivative”. W: *International Journal of Applied Mathematics and Computer Science* 26.3 (2016), s. 533–541.
- [41] K. Oprzedkiewicz. „Non integer order, state space model of heat transfer process using Atangana-Baleanu operator”. W: *Bulletin of the Polish Academy of Sciences. Technical Sciences* 68.1 (2020), s. 43–50.
- [42] L. Torres, J. F. Gomez i R. F. Escobar. *Fractional Derivatives with Mittag-Leffler Kernel Trends and Applications in Science and Engineering*. Berlin: Springer, 2019.
- [43] Q. Yang, F. Liu i I. Turner. „Numerical methods for fractional partial differential equations with Riesz space fractional derivatives”. W: *Applied Mathematical Modelling* 34.1 (2010), s. 200–218.
- [44] T. Kaczorek. „Reachability of cone fractional continuous time linear systems”. W: *International Journal of Applied Mathematics and Computer Science* 19.1 (2009), s. 89–93.
- [45] T. Kaczorek. „Singular fractional linear systems and electrical circuits”. W: *International Journal of Applied Mathematics and Computer Science* 21.2 (2011), s. 379–384.
- [46] K. Balachandran i J. Kokila. „On the controllability of fractional dynamical systems”. W: *International Journal of Applied Mathematics and Computer Science* 22.3 (2012), s. 523–531.

- [47] A. Charef i in. „Fractional System as Represented by Singularity Function”. W: *IEEE Trans Aut. Control* 37.9 (1992), s. 1465–1470.
- [48] A. Douambi, A. Charef i A. V. Besancon. „Approximation and synthesis of non integer order systems”. W: *2nd IFAC Workshop on Fractional Differentiation and its Applications, FDA 2006, Porto, Portugal* (2006).
- [49] A. Douambi, A. Charef i A. V. Besancon. „Optimal approximation, simulation and analog realization of the fundamental fractional order transfer function”. W: *Int. J. Appl. Math. Comp. Sci.* 17.4 (2007), s. 455–462.
- [50] C. Monje i in. *Fractional-Order Systems and Controls: Fundamentals and Applications. Advances in Industrial Control, vol. XXVI*. Springer, London, 2010.
- [51] Oustaloup A. i in. „Frequency-band complex noninteger differentiator: characterization and synthesis”. W: *IEEE Transactions on Circuits and Systems I: Fundamental Theory Applications* 47.1 (2000), s. 25–39.
- [52] B. M. Vinagre i in. „Some Approximations of fractional order operators used in Control Theory and applications”. W: *Fractional Calculus and Applied Analysis* 3.3 (2000), s. 231–248.
- [53] F. Merikh-Bayat. „Rules for selecting the parameters of Oustaloup recursive approximation for the simulation of linear feedback systems containing $PI^\lambda D^\mu$ controller”. W: *Commun. Nonlinear Sci. Numer. Simul.* 17 (2012), s. 1852–1861.
- [54] Y.Q. Chen i K.L. Moore. „Discretization schemes for fractional-order differentiators and integrators”. W: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 49.3 (2002), s. 363–367.
- [55] I. Petras. „Fractional order feedback control of a DC motor”. W: *Journal of Electrical Engineering* 60.3 (2009), s. 117–128.
- [56] R. Stanisławski, W. Huneck i K. Latawiec. „Finite approximations of a discrete-time fractional derivative”. W: *16th International Conference on Methods and Models in Automation and Robotics* (2011).
- [57] I. Petras. „<https://igor.podlubny.website.tuke.sk/USU/matlab/petras/dfod1.m>”. W: ().
- [58] I. Petras. „<https://igor.podlubny.website.tuke.sk/USU/matlab/petras/dfod2.m>”. W: ().
- [59] K. Oprzedkiewicz i W. Mitkowski. „Fractional order state space models of the one-dimensional heat transfer process”. W: *Fractional dynamical systems: methods, algorithms and applications* (2022), s. 361–397.

- [60] K. Oprzedkiewicz. „Positivity problem for the one dimensional heat transfer process”. W: *ISA Transactions* (2021), s. 281–291.
- [61] K. Oprzedkiewicz. „Fractional order, discrete model of heat transfer process using time and spatial Grünwald-Letnikov operator”. W: *Bulletin of the Polish Academy of Sciences. Technical Sciences* 69.1 (2021), s. 1–10.
- [62] K. Oprzedkiewicz. „The interval parabolic system”. W: *Archives of Control Sciences* 13.4 (2003), s. 415–430.
- [63] K. Oprzedkiewicz. „An observability problem for a class of uncertain-parameter linear dynamic systems”. W: *International Journal of Applied Mathematics and Computer Science* 15.3 (2005), s. 331–338.
- [64] W. Mitkowski. *Stabilizacja systemów dynamicznych*. Warszawa: WNT, 1991.
- [65] K. Oprzedkiewicz. „Przykład identyfikacji obiektu parabolicznego”. W: *Elektrotechnika* 16.2 (1997), s. 99–106.
- [66] K. P. Evans i N. Jacob. „Feller Semigroups Obtained by Variable Order Subordination”. W: *Revista Matematica Complutense* 20.2 (2007), s. 293–307.
- [67] A. Pazy. *Semigroups of Linear Operators and Applications to Partial Differential Equations*. Berlin: Springer, 1992.
- [68] K. Oprzedkiewicz. „A controllability problem for a class of uncertain parameters linear dynamic systems”. W: *Archives of Control Sciences* 14.1 (2004), s. 85–100.
- [69] K. Oprzedkiewicz. „A Strejc model-based, semi-fractional (SSF) transfer function model”. W: *Automatics; AGH UST* 16.2 (2012), s. 145–154.
- [70] B. J. Szekeres i F. Izsak. *Numerical solution of fractional order diffusion problems with Neumann boundary conditions, preprint, arXiv:1411.1596 [math.NA]*. 2014.
- [71] S. L. LeVarge. *Semigroups of Linear Operators*. 2003.
- [72] C. Gal i M. Warma. „Elliptic and parabolic equations with fractional diffusion and dynamic boundary conditions”. W: *Evolution Equations and Control Theory* 5.1 (20016), s. 61–103.
- [73] K. Oprzedkiewicz. *wykłady z programowania PLC*. PWSZ w Tarnowie, obecnie ANS w Tarnowie.
- [74] J. Zabczyk. *Zarys matematycznej teorii sterowania*. Warszawa: PWN, 1991.
- [75] R. Isermann i M. Muenchhof. *Identification of Dynamic Systems. An Introduction with Applications*. Springer, Heidelberg, 2011.

- [76] K. Oprzedkiewicz i T. Kołacz. „A non integer order model of frequency speed control in AC motor”. W: *Challenges in automation, robotics and measurement techniques : proceedings of AUTOMATION 2016, Warszawa, Poland* 440 (2016), 287–298.
- [77] D. Mozyrska i E. Pawluszewicz. „Fractional discrete-time linear control systems with initialisation”. W: *International Journal of Control* 1.1 (2011), s. 1–7.
- [78] T. Kaczorek. „Practical stability of positive fractional discrete-time systems”. W: *IBulletin of the Polish Academy of Science: Technical Sciences* 56.4 (2008), s. 313–317.
- [79] M. Buslowicz i T. Kaczorek. „Simple conditions for practical stability of positive fractional discrete-time linear systems”. W: *International Journal of Applied Mathematics and Computer Science* 19.2 (2009), s. 263–269.
- [80] A. Ruszewski. „Practical and asymptotic stability of fractional discrete-time scalar systems described by a new model”. W: *Archives of Control Sciences* 26.4 (2016), s. 441–452.
- [81] K. Oprzedkiewicz. „Approximation method for a fractional order transfer function with zero and pole”. W: *Archives of Control Sciences* 24.41 (2014), s. 409–425.
- [82] R. W. Lewis. „Programming industrial control systems using IEC 1131-3”. W: *IEE* (1998).
- [83] K. H. John i M. Tiegelkamp. *IEC 61131-3: Programming Industrial Automation Systems. Concepts and Programming Languages, Requirements for Programming Systems, Decision Making Aids*. Springer, 2010.
- [84] G. Frey i L. Litz. „Formal methods in PLC programming”. W: *Proc. of the IEEE Conf. on Systems Man and Cybernetics SMC 2000, Nashville, Oct 8-11* (2000), s. 2431–2436.
- [85] W. Mitkowski i K. Oprzedkiewicz. „Optimal sample time estimation for the finite - dimensional discrete dynamic compensator implemented at the “soft PLC” platform”. W: *23rd IFIP TC 7 conference on System modelling and optimization: Cracow, Poland, July 23–27, 2007* (2007), s. 77–78.
- [86] Das S. i Pan I. *Intelligent Fractional Order Systems and Control. An Introduction*. Springer, 2013.
- [87] I. Petras. „Tuning and implementation methods for fractional-order controllers”. W: *Fractional Calculus and Applied Analysis* 15.2 (2012).
- [88] P. Ostalczyk. *Discrete Fractional Calculus. Applications in control and image processing, Series in Computer Vision, vol. 4*. World Scientific Publishing, 2016.

- [89] I. Petras. „Realization of fractional order controller based on PLC and its utilization to temperature control”. W: *Transfer inovácií* 14 (2009).
- [90] J. Kwiecień i B. Filipowicz. „Algorytmy stadne w problemach optymalizacji”. W: *Pomiary Automatyka Robotyka* 12 (2011), s. 152–157.
- [91] J. Kwiecień. „Algorytmy stadne w rozwiązywaniu wybranych zagadnień optymalizacji dyskretnej i kombinatorycznej”. Prac. dokt. Kraków: Akademia Górniczo-Hutnicza, 2015.
- [92] T. Witkowski i P. Antczak. *Algorytmy rojowe w harmonogramowaniu procesów produkcyjnych*. Warszawa: Polskie Wydawnictwo Ekonomiczne, 2022.
- [93] K. Oprzedkiewicz i K. Dziedzic. „PSO identification for discrete fractional order model of heat transfer process”. W: *Advances in non-integer order calculus and its applications : proceedings of the 10th international conference on Non-integer order calculus and its applications* 559 (2020), s. 240–249.
- [94] K. Dziedzic. „Identification of Fractional Order Transfer Function Model Using Biologically Inspired Algorithms”. W: *Advances in Intelligent Systems and Computing* 920 (2019).