

**POLITECHNIKA OPOLSKA
WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI
I INFORMATYKI**

mgr inż. Damian Raczyński

**Redukcja modeli obiektów sterowania
z zastosowaniem obliczeń równoległych
na przykładzie modelu kotła energetycznego**

ROZPRAWA DOKTORSKA

promotor:
dr hab. inż. Włodzimierz Stanisławski, prof. PO

OPOLE 2013

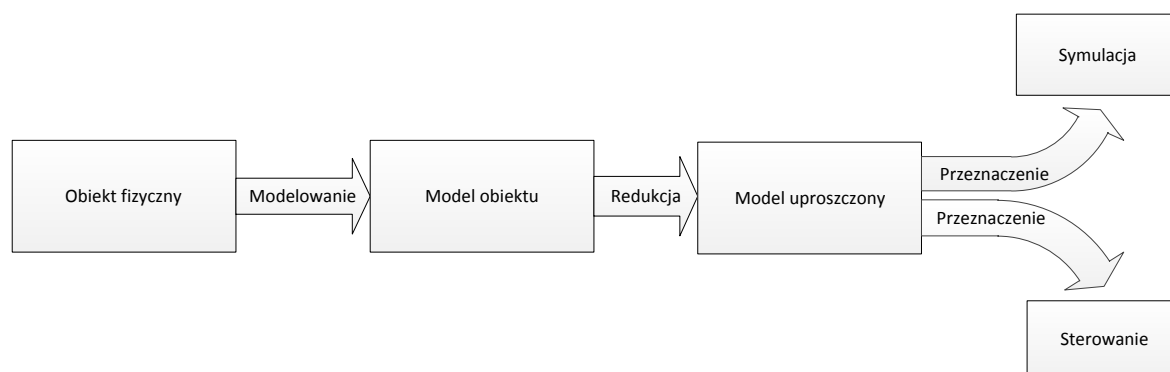
SPIS TREŚCI

1. WSTĘP	4
1.1. METODY REDUKCJI Z ZASTOSOWANIEM SVD.....	5
1.2. DEKOMPOZYCJA MODELU	6
1.3. REDUKCJA POŁĄCZONA Z DEKOMPOZYCJĄ.....	8
1.4. OBLICZENIA RÓWNOLEGLE.....	9
1.5. CEL, TEZA I ZAKRES PRACY	10
2. REDUKCJA RZĘDU MODELI LTI Z WYKORZYSTANIEM SVD	12
2.1. GRAMIANY STEROWALNOŚCI I OBSERWOWALNOŚCI	12
2.1.1. Definicja gramianów.....	12
2.1.2. Metody wyznaczania gramianów	13
2.2. REDUKCJA MODELI Z WYKORZYSTANIEM METODY BTA.....	14
2.3. ALGORYTMY WYZNACZANIA MACIERZY TRANSFORMACJI	15
2.3.1. Metoda wyznaczania dekompozycji Cholesky'ego z nieznacznym przesunięciem wartości własnych	16
2.4. REDUKCJA MODELI Z WYKORZYSTANIEM METODY CZĘSTOTLIWOŚCIOWYCH FUNKCJI WAGOWYCH (ANG. FREQUENCY WEIGHTED).....	16
2.5. UWAGI.....	19
3. METODY JEDNOPOZIOMOWEJ DEKOMPOZYCJI MODELI LTI	21
3.1. DEKOMPOZYCJA FIZYCZNA (ANG. INTERCONNECTED SYSTEMS).....	21
3.2. DEKOMPOZYCJA STRUKTURALNA (ANG. COUPLED SYSTEMS).....	21
3.2.1. Dekompozycja strukturalna uproszczona (<i>StrUp</i>).....	22
3.2.2. Dekompozycja strukturalna (<i>Str</i>).....	23
3.3. DEKOMPOZYCJA MODELI LTI ZE WZGLĘDU NA WARTOŚCI WŁASNE (ANG. SLOW-FAST).....	24
3.4. DEKOMPOZYCJA EPSILON.....	25
3.5. DEKOMPOZYCJA EPSILON SLOW-FAST (<i>ESF</i>)	26
3.6. DEKOMPOZYCJA MULTI-LEVEL SUBSTRUCTURING (MLS)	31
3.7. DEKOMPOZYCJA OPARTA NA DOPEŁNIENIU SCHUR'A – <i>SCHCOMP</i> (ANG. SCHUR COMPLEMENT)	32
3.8. ZMODYFIKOWANA DEKOMPOZYCJA OPARTA NA DOPEŁNIENIU SCHUR'A (<i>SCHCOMPZM</i>)	33
3.9. DEKOMPOZYCJA SCHUR'A (<i>SCH</i>).....	34
3.10. DEKOMPOZYCJA SCHUR-SYLVESTER (<i>SCHSYLV</i>).....	35
3.11. UWAGI.....	36
4. METODY WIELOPOZIOMOWEJ DEKOMPOZYCJI MODELI LTI	38
4.1. DEKOMPOZYCJA WIELOPOZIOMOWA	38
4.2. REALIZACJA DEKOMPOZYCJI WIELOPOZIOMOWEJ Z REDUKCJĄ MODELU	39
4.2.1. Dekompozycja slow-fast na pierwszym poziomie dekompozycji	41
4.2.2. Dekompozycja Schur-Sylvester	41
4.2.3. Dekompozycja Schur'a.....	43
4.2.4. Dekompozycja Schur Complement.....	44
4.2.5. Dwupoziomowa dekompozycja Schur complement	45
4.2.6. Dekompozycja strukturalna i strukturalna uproszczona	48
4.2.7. Zmodyfikowana dekompozycja Schur complement	48
4.3. UWAGI.....	50
5. REALIZACJA ALGORYTMÓW REDUKCJI MODELI LTI Z ZASTOSOWANIEM OBLICZEŃ RÓWNOLEGLYCH	51
5.1. SPOSOBY ZRÓWNOLEGLANIA PROCESU OBLICZENIOWEGO	51
5.2. ARCHITEKTURA FERMI PROCESORÓW GRAFICZNYCH FIRMY NVIDIA	52
5.3. CHARAKTERYSTYKA ŚRODOWISKA OBLICZENIOWEGO MATLAB	54
5.4. WYKORZYSTYWANE BIBLIOTEKI OBLICZENIOWE	56
5.5. ZAIMPLEMENTOWANA APLIKACJA OBLICZENIOWA	57
5.6. CHARAKTERYSTYKA ŚRODOWISKA OBLICZENIOWEGO	63
5.7. UWAGI.....	65

6. MODELE MATEMATYCZNE PODSYSTEMÓW KOTŁA ENERGETYCZNEGO BP-1150 I ICH DEKOMPOZYCJA	67
6.1. KOCIOŁ ENERGETYCZNY BP-1150.....	67
6.2. DEKOMPOZYCJA FIZYCZNA MODELU MATEMATYCZNEGO PAROWNIKA KOTŁA BP-1150	70
6.3. MODELE MATEMATYCZNE PODSYSTEMÓW PAROWNIKA.....	73
6.3.1. Rury ekranowe	73
6.3.2. Separator.....	79
6.4. MODELE ZLINEARYZOWANE.....	80
6.5. UWAGI.....	81
7. ZREDUKOWANE MODELE PODSYSTEMÓW KOTŁA ENERGETYCZNEGO BP-1150	83
7.1. WYBÓR METOD DEKOMPOZYCJI DLA MODELI PODSYSTEMÓW PAROWNIKA.....	83
7.1.1. Analiza błędów oraz stabilności zredukowanych modeli dla dekompozycji jednowymiarowej	84
7.1.2. Analiza błędów oraz stabilności zredukowanych modeli dla dekompozycji dwuwymiarowej.....	87
7.2. MODELE ZREDUKOWANE PODSYSTEMÓW RUR EKRANOWYCH.....	91
7.2.1. Wybór metod dekompozycji.....	91
7.2.2. Parametry redukcji.....	92
7.2.3. Modele zredukowane strefy dogrzewu	92
7.2.4. Modele zredukowane strefy odparowania I	95
7.2.5. Modele zredukowane strefy odparowania II	98
7.2.6. Modele zredukowane strefy odparowania III.....	100
7.3. MODELE ZREDUKOWANE SEPARATORA	103
7.4. MODELE ZREDUKOWANE UKŁADU CYRKULACJI	105
7.5. POWTÓRNIENIE ZREDUKOWANYCH MODELI RUR EKRANOWYCH	106
7.6. MODELE ZREDUKOWANE PAROWNIKA	108
7.7. UWAGI.....	111
8. WERYFIKACJA MODELI ZREDUKOWANYCH	113
8.1. UKŁAD REGULACJI CIŚNIENIA W KOTLE BP-1150.....	113
8.2. UKŁAD REGULACJI	116
8.3. REGULATOR LQG	116
8.4. PROJEKT REGULATORA NA PODSTAWIE ZREDUKOWANEGO MODELU PAROWNIKA.....	117
8.5. UWAGI.....	122
PODSUMOWANIE	124
LITERATURA	129
SPIS ILUSTRACJI.....	136
SPIS TABEL	141
DODATEK A. ZBIÓR WYBRANYCH PROCEDUR LAPACK I CULA	142
DODATEK B. WYBRANE FUNKCJE I PROCEDURY MATLAB ZWIĄZANE Z DEKOMPOZYCJĄ I REDUKCJĄ RÓWNOLEGŁĄ	143
DODATEK C. WYBRANE FUNKCJE ŚRODOWISKA MATLAB ZWIĄZANE Z WYZNACZANIEM ROZWIĄZANIA RÓWNIANIA LAPUNOWA	157
DODATEK D. WYBRANE FUNKCJE I PROCEDURY JĘZYKA C++ ZWIĄZANE Z REALIZACJĄ OBLICZEŃ Z WYKORZYSTANIEM PROCESORA GRAFICZNEGO.....	157
DODATEK E. WYBRANE FUNKCJE I PROCEDURY JĘZYKA C++ ZWIĄZANE Z WYZNACZENIEM ROZWIĄZANIA RÓWNIANIA LAPUNOWA W ŚRODOWISKU PROCESORA GRAFICZNEGO.....	160
DODATEK F. WYBRANE FUNKCJE I PROCEDURY JĘZYKA C++ ZWIĄZANE Z REDUKCJĄ SVD W ŚRODOWISKU PROCESORA GRAFICZNEGO	161
DODATEK G. – WYKRESY BŁĘDÓW I MAPY STABILNOŚCI DLA REDUKCJI POWIĄZANEJ Z DEKOMPOZYCJĄ JEDNOWYMIAROWĄ.....	162
DODATEK H. – WYKRESY BŁĘDÓW I MAPY STABILNOŚCI DLA REDUKCJI POWIĄZANEJ Z DEKOMPOZYCJĄ DWUWYMIAROWĄ.....	165

1. Wstęp

Postęp w dziedzinie informatyki wpływa na szybki rozwój oprogramowania do modelowania złożonych obiektów dynamicznych, a szczególnie obiektów o czasoprzestrzennej dynamice. Do modelowania obiektów tego typu stosuje się metodę elementów skończonych. Wraz ze wzrostem mocy obliczeniowej komputerów pojawił się cały szereg aplikacji, takich jak: ANSYS [1], ABAQUS [2], NASTRAN [3], COSMOS [4], ALGOR [5], PATRAN [6], HyperMesh [7], które umożliwiają modelowanie skomplikowanych procesów z wykorzystaniem metody elementów skończonych. Wysoka dokładność modelu wynikowego wymaga użycia dużej liczby elementów skończonych - dyskretyzacja kontinuum prowadzi do złożonego układu równań różniczkowych zwyczajnych. Bardzo często uzyskane równania modelujące dynamikę danego procesu są na tyle skomplikowane, że wyznaczenie ich analitycznego rozwiązania jest niemożliwe lub niepraktyczne [8]. Analiza właściwości otrzymanego modelu lub wyznaczenie na jego podstawie układu sterowania jest praktycznie niemożliwa, nawet z wykorzystaniem najnowszych technologii [9]. Z tego powodu rozwijane są algorytmy redukcji złożonych modeli dynamicznych [10] [11]. Uzyskany model uproszczony, którego rozmiar jest dostosowany do ograniczeń mocy obliczeniowej, dokładności oraz pojemności pamięci, posiadający główne cechy modelu oryginalnego, używany jest w miejsce modelu oryginalnego (rys.1) [10].



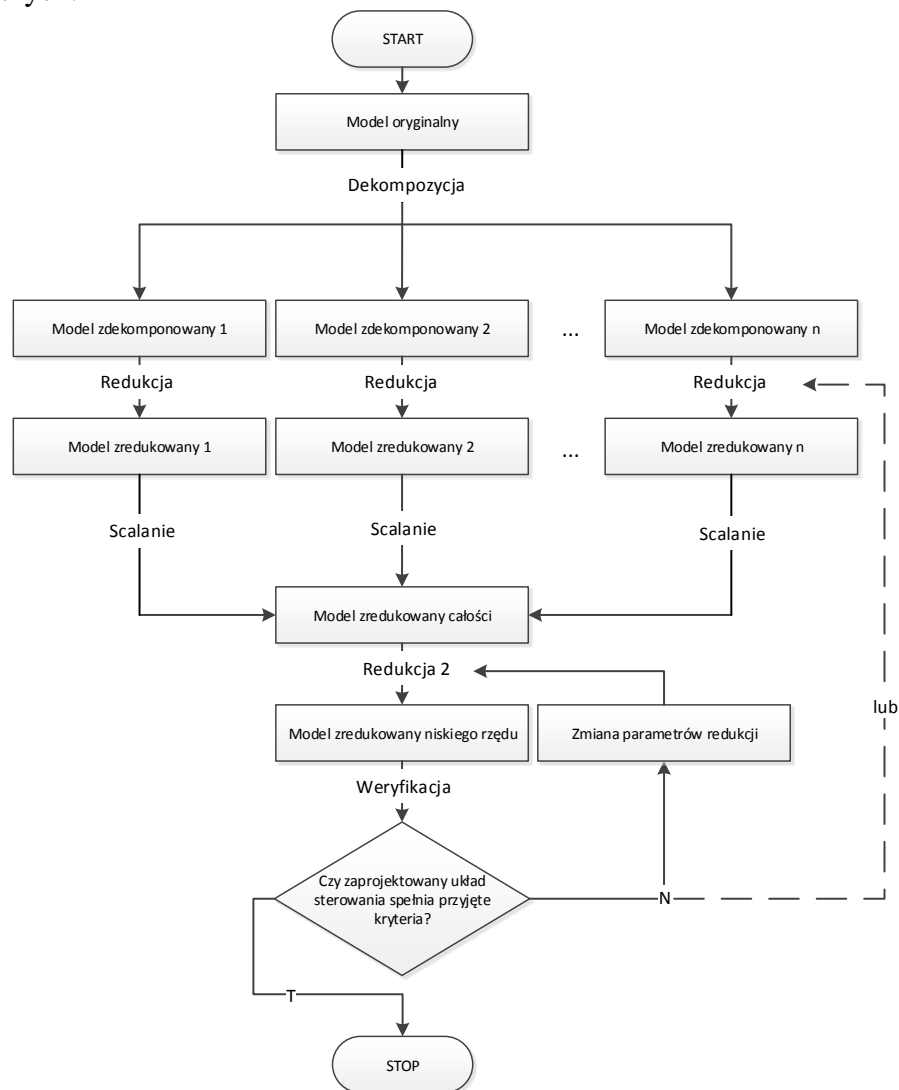
RYSUNEK 1. PROCES REDUKCJI WYMIARU MODELI OBIEKTÓW FIZYCZNYCH

W przypadku, gdy modele oryginalne posiadają znaczny rozmiar, ich redukcja z wykorzystaniem metod SVD, o złożoności obliczeniowej rzędu $O(n^3)$ [12] [10], staje się procesem długotrwałym. Dodatkowo wykorzystanie metody częstotliwościowych funkcji wagowych [13] (ang. Frequency Weighted) wraz z ewolucyjnym algorytmem doboru optymalnych parametrów filtrów eAMOR [14] prowadzi do konieczności realizacji wielu setek pojedynczych redukcji SVD dla modelu oryginalnego z dołączonymi filtrami o różnych parametrach.

Autor w pracy przedstawia możliwą do zastosowania metodologię umożliwiającą realizację redukcji złożonego modelu z wykorzystaniem metody częstotliwościowych funkcji wagowych. Schemat blokowy realizowanych operacji w celu uzyskania modelu zredukowanego został przedstawiony na rys. 2.

Złożoność procesu obliczeniowego zostaje ograniczona poprzez zastosowanie metod dekompozycyjnych, natomiast uzyskane w wyniku dekompozycji modele poddawane są procesowi redukcji z wykorzystaniem środowiska klastrowego. Dla zobrazowania poprawności opracowanej metodologii rozpatrywany jest przypadek redukcji modelu obiektu fizycznego - parownika kotła energetycznego BP-1150. Wyniki redukcji są następnie

weryfikowane poprzez projekt układu sterowania na podstawie wyników modeli zredukowanych.



RYSUNEK 2. SCHEMAT BLOKOWY REALIZACJI PROCESU REDUKCJI MODELI ZŁOŻONYCH Z WYKORZYSTANIEM OPRACOWANEJ METODOLOGII

1.1. Metody redukcji z zastosowaniem SVD

Najczęściej stosowane metody redukcji modeli LTI można podzielić na trzy grupy [15]:

- metody wykorzystujące dekompozycję SVD,
- metody związane z rozwinięciem transmitancji modelu w szereg Laurenta wokół jednego lub kilku punktów – metody momentów. Metody te posiadają mniejszą złożoność obliczeniową, lecz nie gwarantują stabilności modeli zredukowanych.
- metody SVD-Kryłowa, bazujące na wyznaczaniu aproksymacji gramianów sterowalności i obserwowalności.

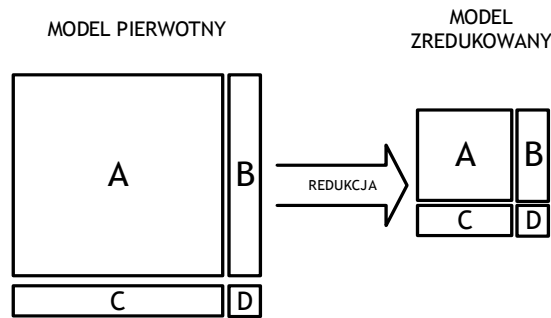
W pracy zastosowano metody redukcji modeli zapisanych w postaci równań macierzowych w przestrzeni stanów (1) z wykorzystaniem metod opartych na dekompozycji SVD [16].

$$\Sigma : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \quad (1)$$

gdzie:

A - macierz stanu, B - macierz wejść, C - macierz wyjść, D - macierz transmisyjna [17].

W odniesieniu to takich modeli proces redukcji związany jest ze zmniejszeniem liczby wierszy i/lub kolumn macierzy A, B, C z równania (1) (rys. 3).



RYSUNEK 3. OPERACJA REDUKCJI RZĘDU MODELI W PRZESTRZENI STANÓW

Do niewątpliwych zalet redukcji z zastosowaniem SVD należy zaliczyć gwarancję stabilności modelu zredukowanego [18] oraz możliwość oszacowania maksymalnego błędu aproksymacji [19] [20]. Wadą redukcji opartej na SVD jest znaczna złożoność obliczeniowa ($O(n^3)$), która ogranicza przydatność klasycznych metod redukcji SVD do modeli o niewielkim rzędzie [12]. Do podstawowych algorytmów redukcji modeli z wykorzystaniem dekompozycji SVD należy zaliczyć metody:

- **BTA** (*ang. Balanced Truncation Approximation*) – metoda bazuje na zrównoważonej realizacji modelu. Wykorzystuje gramiany sterowalności i obserwowalności w celu usunięcia z modelu oryginalnego części zmiennych stanu [20].
- **SPA** (*ang. Singular Perturbation Approximation*) – podobnie jak BTA bazuje na zrównoważonej realizacji modelu. Wprowadza modyfikacje mające na celu poprawę dokładności modelu zredukowanego dla zakresu niższych częstotliwości [21].
- **FW** (*ang. Frequency Weighted*) – wymaga doboru odpowiednich częstotliwościowych funkcji wagowych na wejściach i wyjściach modelu oryginalnego w celu wyznaczenia zmodyfikowanych gramianów sterowalności i obserwowalności. Takie podejście ma na celu poprawę dokładności modelu zredukowanego dla określonego zakresu częstotliwości [22].
- **FD** (*ang. Frequency Domain*) – wykorzystuje gramiany sterowalności i obserwowalności wyznaczone w dziedzinie częstotliwościowej [23].
- **TLBR** (*ang. Time-Limited Balanced Reduction*) – wykorzystuje gramiany sterowalności i obserwowalności wyznaczone dla określonego zakresu czasowego [24].
- **HNA** (*ang. Hankel Norm Approximation*) – wykorzystuje normę Hankela w celu minimalizacji błędu aproksymacji [20].
- **FWSPA** (*ang. Frequency Weighted Singular Perturbation Approximation*) – wymaga doboru funkcji wagowych w odniesieniu do metody **SPA** [25].
- **FWHNA** (*ang. Frequency Weighted Hankel Norm Approximation*) – wymaga doboru funkcji wagowych w odniesieniu do metody **HNA** [25].

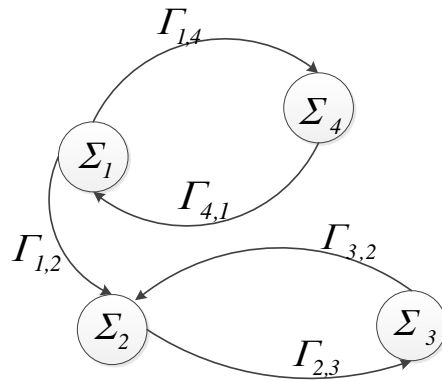
W rozdziale 2 przedstawiono szerzej metody **BTA** oraz **FW**.

1.2. Dekompozycja modelu

Dekompozycja modelu bazuje na wyodrębnieniu w jego strukturze części, które poprzez wzajemne relacje między sobą (wzajemne oddziaływania) tworzą obiekt złożony oraz zastosowaniu przekształceń umożliwiających wydzielenie wyodrębnionych podsystemów.

Model obiektu złożonego, zawierający zbiór modeli podsystemów (*ang. coupled systems*), może zostać przedstawiony za pomocą grafu skierowanego. Na rys. 4

przedstawiono przykładowy system Σ składający się z 4 podsystemów powiązanych ze sobą.



RYSUNEK 4. GRAFOWA REPREZENTACJA SYSTEMÓW POWIĄZANYCH

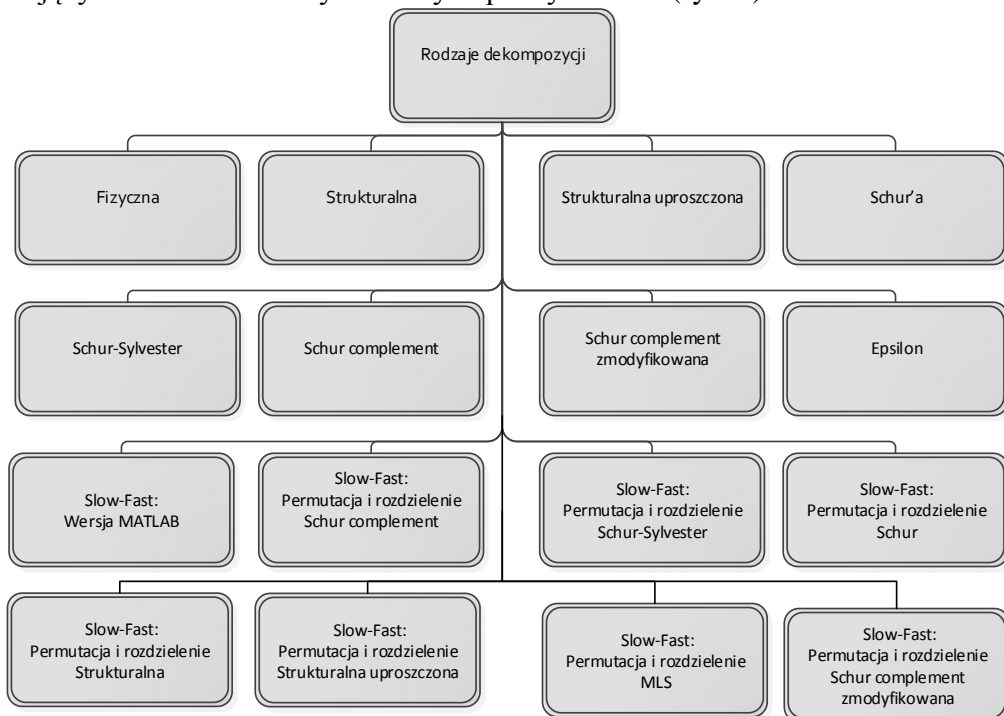
A_1	$\Gamma_{2,1}$	$\Gamma_{3,1}$	$\Gamma_{4,1}$
$\Gamma_{1,2}$	A_2	$\Gamma_{3,2}$	$\Gamma_{4,2}$
$\Gamma_{1,3}$	$\Gamma_{2,3}$	A_3	$\Gamma_{4,3}$
$\Gamma_{1,4}$	$\Gamma_{2,4}$	$\Gamma_{3,4}$	A_4

RYSUNEK 5. STRUKTURA MACIERZY STANU A SYSTEMU, W KTÓRYM WYDZIELONO CZTERY PODSYSTEMY

Połączenia między podsystemami nazywane są interfejsami. Oznaczenie interfejsu $\Gamma_{n,m}$ wskazuje, że system n wpływa na zmienne stanu systemu m . Połączenia między poszczególnymi podsystemami realizowane są poprzez niezerowe wartości obecne w blokach nie leżących na diagonalu w macierzy stanu A . Dla systemu przedstawionego na rys. 4, struktura macierzy A może mieć postać przedstawioną na rys. 5.

Obecność niezerowych wartości w danym interfejsie oznaczono na rysunku kropkami. Na diagonalu macierzy A systemu Σ znajdują się kwadratowe bloki, będące macierzami stanu A_1, A_2, A_3, A_4 poszczególnych podsystemów. Analogicznie do struktury macierzy A systemu Σ przeprowadza się podział macierzy wejść, wyjść i wektora zmiennych stanu.

Autor w treści pracy zaproponował zastosowanie szeregu metod dekompozycyjnych umożliwiających rozdzielenie wyróżnionych podsystemów (rys. 6).



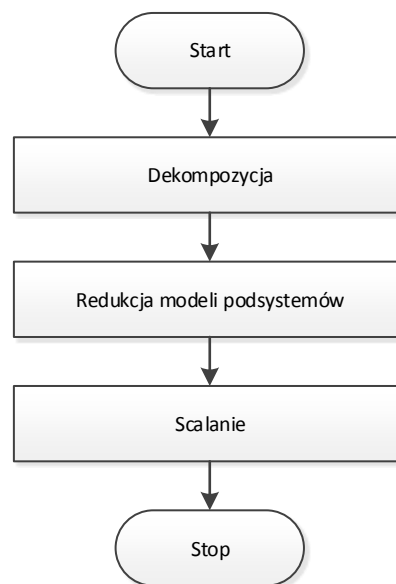
RYSUNEK 6. WYKORZYSTYWANE DEKOMPOZYCJE MODELI LTI

Szczegółowy opis poszczególnych metod dekompozycji został zamieszczony w rozdziale 3.

1.3. Redukcja połączona z dekompozycją

Redukcja z wykorzystaniem SVD charakteryzuje się złożonością obliczeniową $O(n^3)$, gdzie n określa rząd modelu redukowanego. Dekompozycja redukowanego modelu na m podsystemów, przy założeniu równego podziału, powoduje zmniejszenie złożoności obliczeniowej do poziomu $O\left(m\left(\frac{n}{m}\right)^3\right)$. Dodatkowo dekompozycja modelu na szereg części umożliwia ich równoległą redukcję z wykorzystaniem systemu klastrowego. Każdy wyróżniony podsystem może być osobnym zadaniem redukcji realizowanym na osobnym węźle klastra.

Ogólny schemat redukcji modeli połączonej z dekompozycją przedstawia rys. 7.

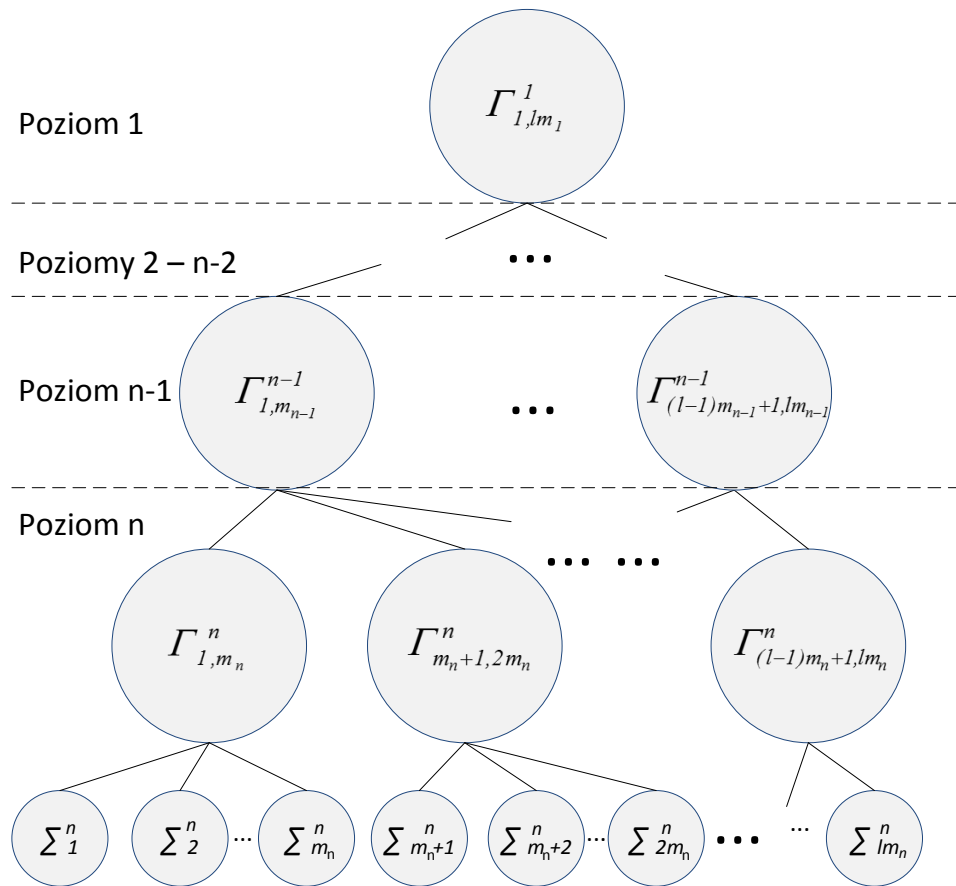


RYSUNEK 7. SCHEMAT BLOKOWY REDUKCJI POŁĄCZONEJ Z DEKOMPOZYCJĄ

Realizacja wielopoziomowej dekompozycji związana jest z wyodrębnianiem modeli składowych w strukturze modelu nadrzędnego, z wykorzystaniem wybranego algorytmu, a następnie dekompozycją modeli podsystemów, aż do osiągnięcia pożądanego rozmiaru modeli znajdujących się na najniższym poziomie dekompozycji. Algorytmy dekompozycji wykorzystane na poszczególnych poziomach dekompozycji nie muszą być te same, należy jednak zaznaczyć, że niektóre ich kombinacje dla pewnych przypadków modeli mogą powodować niską jakość modeli wynikowych. Oznaczając równania stanu opisujące j -ty model wyznaczony na i -tym poziomie dekompozycji poprzez:

$$\sum_j^i : \begin{cases} E_j^i \dot{x}_j^i(t) = A_j^i x_j^i(t) + B_j^i u(t) \\ y_j^i(t) = C_j^i x_j^i(t) \end{cases} \quad (2)$$

oraz interfejs łączący modele $\sum_d^i \dots \sum_p^i : \Gamma_{d,p}^i$, n poziomowa dekompozycja, w której na każdym poziomie wyróżniono $l_i \cdot m_i$ (l_i - liczba interfejsów na i 'tym poziomie, m_i - liczba modeli połączonych jednym interfejsem na i 'tym poziomie) modeli może zostać przedstawiona w postaci struktury grafu drzewiastego (rys. 8).



RYSUNEK 8. GRAF DEKOMPOZYCJI WIELOPOZIOMOWEJ

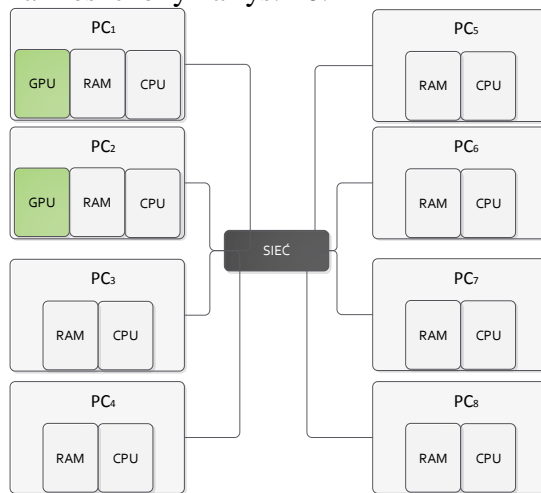
Realizacja redukcji polega na wyznaczeniu modeli zredukowanych dla wszystkich modeli najniższego poziomu oraz zastosowaniu uzyskanych macierzy transformacji do przekształcenia interfejsów:

1.4. Obliczenia równoległe

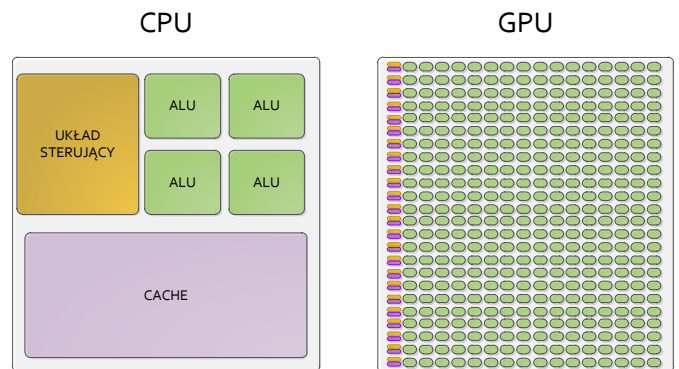
Architektury współczesnych komputerów cechują się szeregiem rozwiązań technicznych, takich jak wielordzeniowość, pamięć podręczna, jednostki wektorowe, umożliwiającą przyspieszenie procesu obliczeniowego [26] [27] [28]. W celu realizacji masywnego procesu obliczeniowego często pojedyncze komputery łączone są w klastery komputerowe.

Do równoległej realizacji dekompozycji i redukcji modeli LTI wykorzystano środowisko klastrowe, w którego skład wchodzi komputery PC wyposażone w procesory oparte na architekturze x86 (CPU) oraz procesory graficzne oparte na architekturze FERMI (GPU) (rys. 9). Zasadnicze różnice w architekturze obu typów jednostek obliczeniowych wynikają z liczby rdzeni oraz rozmiaru pamięci podręcznej. W odróżnieniu od klasycznych procesorów, procesory graficzne zawierają znaczną liczbę rdzeni, realizujących podstawowe operacje arytmetyczne oraz niewielkiej pamięci podręcznej. Taka architektura, połączona z efektywnym zarządzaniem wątkami wykonującymi się na poszczególnych rdzeniach, gwarantuje wysoką wydajność dla przetwarzania dużych danych blokowych. Procesory graficzne doskonale radzą sobie z wykonywaniem obliczeń macierzowych lub wektorowych o znacznej liczbie elementów. Należy jednak zaznaczyć, że w przypadku częstej wymiany danych pomiędzy pamięcią główną a pamięcią karty graficznej proces obliczeniowy może zostać w znacznym stopniu spowolniony. W przypadku, gdy wielkość przetwarzanych danych nie jest wystarczająco duża, lub gdy realizowany algorytm wymaga częstych odwołań

do pamięci głównej, bardziej efektywnym rozwiązaniem jest wykonywanie obliczeń z zastosowaniem CPU, który lepiej sprawdza się przy przetwarzaniu programu sekwencyjnego [29]. Ogólny schemat porównujący oba typy jednostek obliczeniowych został zamieszczony na rys. 10.



RYSUNEK 9. STRUKTURA ZASTOSOWANEGO W PRACY KLASTRA OBLICZENIOWEGO



RYSUNEK 10. PORÓWNANIE ARCHITEKTUR CPU I GPU

1.5. Cel, teza i zakres pracy

Celem pracy jest:

- Opracowanie algorytmów oraz programów do redukcji rzędu złożonych modeli LTI z zastosowaniem obliczeń równoległych.
- Wykorzystanie opracowanych programów do redukcji rzędu modelu parownika kotła energetycznego BP-1150.
- Weryfikacja opracowanych metod redukcji oraz zredukowanych modeli parownika kotła energetycznego BP-1150.

Aby osiągnąć cel pracy, należy rozwiązać następujące zadania:

1. Opracowanie algorytmów oraz programów dla celów dekompozycji i redukcji rzędu modeli LTI dla środowiska MATLAB oraz procesorów graficznych NVIDIA z wykorzystaniem metody FW.
2. Dekompozycja oraz redukcja rzędu modeli podsystemów parownika kotła energetycznego BP-1150.
3. Zastosowanie modeli zredukowanych do zaprojektowania układu regulacji parownika kotła energetycznego BP-1150.
4. Ocena uzyskanych modeli zredukowanych na podstawie analizy właściwości zaprojektowanego układu regulacji w zastosowaniu do modelu wysokiego rzędu (wyjściowego).

Teza pracy może zostać sformułowana następująco:

Zastosowanie metod dekompozycji oraz obliczeń równoległych umożliwia skuteczną realizację redukcji rzędu złożonych liniowych stacjonarnych ciągłych modeli obiektów sterowania metodą częstotliwościowych funkcji wagowych.

W rozdziale drugim zamieszczono opis podstawowych pojęć związanych z redukcją z wykorzystaniem SVD. Przedstawiono definicję gramianów sterowalności i obserwowalności oraz ich rolę w procesie redukcji. Porównano metody wyznaczania

macierzy transformacji równoważących model oryginalny a także opisano proces redukcji metodą zrównoważonej realizacji (BTA). Przedstawiono redukcję z użyciem metody częstotliwościowych funkcji wagowych wprowadzoną w pracach D. Ennsa oraz jej modyfikacje.

Rozdział trzeci zawiera opis metod dekompozycji jednopoziomowych modeli LTI. W rozdziale zaproponowano modyfikację algorytmu dekompozycji Epsilon wprowadzonej przez D. Šiljaka [30], jako podstawę dla opracowania szeregu metod wydzielających część szybką i część wolną analizowanych modeli podsystemów parownika kotła energetycznego BP-1150 (dekompozycja slow-fast). Analiza metody MLS, opisanej w pracach F. Blömelinga [31], pozwoliła na opracowanie metod Schur-Complement oraz zmodyfikowanej dekompozycji Schur-Complement. Obie metody uwzględniają w procesie scalania interfejsy łączące podsystemy, w odróżnieniu od metody MLS.

W rozdziale czwartym przedstawiono sposób realizacji dekompozycji dwupoziomowych z wykorzystaniem metod zaprezentowanych w rozdziale trzecim. Opracowana metodologia pozwala na wykonywanie dekompozycji różnego typu dla poziomu pierwszego i drugiego. Przeanalizowano postać interfejsów między podsystemami wynikającą z realizacji różnego typu dekompozycji dwupoziomowych. Rozważania wynikające z treści rozdziału są podstawą realizacji programów redukcji modeli podsystemów parownika. Stosując przyjętą metodologię wyróżniono 78 różnych kombinacji dekompozycji.

Rozdział piąty zawiera charakterystykę środowiska obliczeń równoległych opracowanego na potrzeby redukcji modeli wysokiego rzędu. Warstwa sprzętowa została zrealizowana w oparciu o klaster komputerowy z wielordzeniowymi procesorami x86 oraz procesorami graficznymi. W rozdziale znajduje się krótka charakterystyka architektury procesorów graficznych oraz podstawowe informacje na temat budowy i właściwości klastra. Warstwa oprogramowania, składająca się z szeregu programów napisanych dla środowiska MATLAB oraz C++. W rozdziale znajduje się opis wykorzystywanych bibliotek oraz struktury i funkcjonalności stworzonych programów.

W rozdziale szóstym, na podstawie prac W. Stanisławskiego [32] zostały scharakteryzowane modele podsystemów parownika kotła energetycznego BP-1150. W wyniku dekompozycji fizycznej, wydzielone zostały modele: rur ekranowych (a następnie stref: dogrzewu, odparowania I, odparowania II, odparowania III), separatora, układu cyrkulacji, wykorzystywane w rozdziale siódmym.

Rozdział siódmy zawiera analizę opracowanych metod dekompozycji i redukcji w odniesieniu do modeli podsystemów parownika. Na podstawie modeli niskiego rzędu dokonana zostaje ocena opracowanych metod dekompozycyjnych. Najlepsze jakościowo metody wykorzystywane są kolejno do redukcji modeli podsystemów parownika wysokiego rzędu. W rozdziale przedstawiono analizę błędów i stabilności uzyskanych modeli.

W rozdziale ósmym zawarto weryfikację modeli zredukowanych. Na podstawie modeli niskiego rzędu projektowany został szereg układów sterowania z wykorzystaniem wybranych metod. Uzyskane układy regulacji zostały zastosowane do modelu wysokiego rzędu, po czym następuje ich ocena.

2. Redukcja rzędu modeli LTI z wykorzystaniem SVD

2.1. Gramiany sterowalności i obserwowalności

Redukcja modeli LTI z wykorzystaniem SVD wymaga wyznaczenia macierzy gramianów sterowalności \mathbf{P} i obserwowalności \mathbf{Q} lub ich dekompozycji Choleskyego. [10] Wyznaczanie gramianów jest procesem kosztownym obliczeniowo. Wybór oraz implementacja odpowiedniego algorytmu wyznaczania gramianów sterowalności i obserwowalności mają kluczowe znaczenie dla czasu realizacji operacji redukcji.

2.1.1. Definicja gramianów

Minimalna energia sygnału sterującego, niezbędna do osiągnięcia stanu $x(0)=x_0$ w nieskończonym czasie określona jest zależnością: $\int_{-\infty}^0 u^*(t)u(t)dt = x_0^* P^{-1} x_0$ [33]. Gdzie P oznacza gramian sterowalności. Gramian sterowalności może zostać wyznaczony z zależności [34]:

$$P = \int_0^{\infty} e^{At} B B^* e^{A^*t} dt \quad (3)$$

Dla skończonego czasu t pochodna gramianu sterowalności może zostać zapisana jako [16]:

$$\begin{aligned} \dot{P}(t) &= \int_0^t \frac{d}{dt} [e^{A^*t} B B^* e^{At}] dt + P(0) = \int_0^t [A^* e^{A^*t} B B^* e^{At} + e^{A^*t} B B^* e^{At} A] dt + B B^* = \\ &= A^* \int_0^t e^{A^*t} B B^* e^{At} dt + \int_0^t e^{A^*t} B B^* e^{At} dt A + B B^* = AP + PA^* + B B^* \end{aligned} \quad (4)$$

Dla układu stabilnego i dla czasu nieskończonego, gramian sterowalności dąży do ustalonej wartości, w związku z czym pochodna gramianu sterowalności dąży do zera. Można zatem zapisać [34]:

$$AP + PA^* + B B^* = 0 \quad (5)$$

gdzie (5) jest równaniem Lapunowa.

Energia sygnału wyjściowego, dla stanu początkowego $x(0)=x_0$, przy braku sterowania określona jest zależnością $\int_0^{\infty} y^*(t)y(t)dt = x_0^* Q x_0$ [33]. Gdzie Q oznacza gramian obserwowalności. Gramian obserwowalności można wyznaczyć z zależności [35]:

$$Q = \int_0^{\infty} e^{A^*t} C^* C e^{At} dt \quad (6)$$

Dla skończonego czasu t , pochodna gramianu obserwowalności może zostać wyznaczona następująco [16]:

$$\begin{aligned} \dot{Q}(t) &= \int_0^t \frac{d}{dt} [e^{A^*t} C^* C e^{At}] dt + Q(0) = \int_0^t [A^* e^{A^*t} C^* C e^{At} + e^{A^*t} C^* C e^{At} A] dt + C^* C = \\ &= A^* \int_0^t e^{A^*t} C^* C e^{At} dt + \int_0^t e^{A^*t} C^* C e^{At} dt A + C^* C = A^* Q + Q A + C^* C \end{aligned} \quad (7)$$

Dla układu stabilnego i dla nieskończonego czasu gramian obserwowalności dąży do wartości ustalonej, w związku z czym pochodna gramianu obserwowalności dąży do zera. Można zatem zapisać [34]:

$$A^* Q + Q A + C^* C = 0 \quad (8)$$

Jeżeli dla systemu Σ gramian sterowalności P jest dodatnio określony, to system Σ jest sterowalny, oraz gdy gramian obserwowalności Q , dla systemu Σ jest dodatnio określony, to system Σ jest obserwowalny [36]. Macierze gramianów sterowalności i obserwowalności są macierzami symetrycznymi.

2.1.2. Metody wyznaczania gramianów

W literaturze można znaleźć szereg metod wyznaczania gramianów na podstawie równania Lapunowa:

1. Metoda Kroneckera [37] [38] [39],
2. Metoda Bartels-Stewart'a [40] [41] [42] [43] [44] [45] ,
3. Metoda Smith'a [46] [47] [40] [48],
4. Metoda ADI [49] [50] [51] [52],
5. Metoda Funkcji Znak [53] [54] [55] [56] [57] [58],
6. Metody niskiego rzędu [59] [60],
 - Funkcji znaku niskiego rzędu [61] [62] [63] [64] [65] [66] [67] [68],
 - Smith'a [49] [69], [70],
 - ADI [36] [49] [71] [36] [72] [69] [73].

Metoda Kroneckera należy do bezpośrednich metod rozwiązywania równania Lapunowa. Równanie Lapunowa przekształcone do układu n^2 równań liniowych [74] [37] [38] [39]. Wymiar powstałego układu równań powoduje, że dla modeli wyższego rzędu metoda jest praktycznie nie do zastosowania (ze względu na wymogi dotyczące pamięci operacyjnej). Kolejną metodą bezpośrednią rozwiązywania równania Lapunowa jest algorytm Bartels-Stewart'a. Metoda wykorzystuje dekompozycję Schur'a w celu doprowadzenia macierzy A lub A^* z równania Lapunowa do postaci trójkątnej górnej [40] [41] [42] [43] [44] [45]. Po wyznaczeniu rozwiązania przekształconych równań, wykonana zostaje odwrotna transformacja wyniku z wykorzystaniem ortogonalnych macierzy dekompozycji Schur'a. W algorytmie Smith'a równania Lapunowa dla systemów ciągłych (5) i (8) zostają przekształcone przy użyciu biliniowej transformacji [75] do postaci dyskretnej [46] [47]. Przy założeniu asymptotycznej stabilności macierzy A , kolejne przybliżenia gramianów sterowalności i obserwowalności mogą zostać wyznaczone z zależności iteracyjnej [40] [48] [40] [76]. Istnieje również modyfikacja algorytmu - Smith(1), w której próbuje się uzyskać szybszą zbieżność wyników rozwiązania [10]. Metoda ADI (*ang. Alternating Direction Implicit*) po raz pierwszy została wprowadzona w [77]. Jest ona używana do rozwiązania układu równań liniowych w postaci $My = b$, gdzie macierz M jest symetryczna, dodatnio określona i może zostać podzielona na sumę dwóch symetrycznych dodatnio określonych macierzy $M = M_1 + M_2$. Gdy wspomniane warunki są spełnione, układ równań można rozwiązać stosując proces iteracyjny [78]. W odniesieniu do równań Lapunowa metoda ADI została przedstawiona między innymi w [49] [50]. Głównym problemem związanym z zastosowaniem metody jest konieczność doboru współczynników przesunięć (*ang. ADI shifts*). Można odnaleźć wiele prac na temat doboru optymalnych przesunięć ADI dla modeli z symetryczną macierzą A [79] [34] [36] [80] [51] [52]. W przypadku, gdy macierz A jest niesymetryczna, konieczne jest zastosowanie algorytmów heurystycznych, które próbują wyznaczyć suboptymalne wartości przesunięć. Dużą rolę w tym zakresie odgrywają algorytmy oraz przybornik środowiska MATLAB LYAPACK opracowany przez T. Penzla [81] [51]. Metoda funkcji znaku jest jedną z najczęściej stosowanych metod w odniesieniu do równania Lapunowa o dużych rozmiarach. Najszerszej używaną ze znanych metod wyznaczania wartości funkcji znaku jest iteracyjna metoda Newtona [82]. W odniesieniu do równania Lapunowa często stosowane są metody bazujące na wprowadzeniu parametru zwiększającego szybkość zbieżności rozwiązania [56] [55] [83] [63] [84].

Po raz pierwszy metodę rozwiązania równania Lapunowa w postaci dekompozycji Cholesky'ego zaproponował S. Hammarling [60]. Metody niskiego rzędu wyznaczają macierze S^T dla gramianu sterowalności oraz R^T dla gramianu obserwowalności, gdzie $P = S^T S$, $Q = R^T R$. Są one efektywne dla modeli z niewielką liczbą wejść i wyjść (gdy $\text{rank}(BB^*) \ll n$ oraz $\text{rank}(C^*C) \ll n$). Problem, który się pojawia przy implementacji algorytmów niskiego rzędu dotyczy przyrostu liczby kolumn/wierszy rozwiązania w każdej iteracji. Stosowane rozwiązania polegają na wykorzystaniu dekompozycji QR, LU, RRQR (ang. rank-revealing QR) [85] [86] oraz RRLU (ang. rank-revealing LU) [87] [88] w celu zmniejszenia liczby kolumn/wierszy otrzymanych macierzy.

2.2. Redukcja modeli z wykorzystaniem metody BTA

Metoda zrównoważonej realizacji modelu zredukowanego (ang. Balanced Truncation Approximation), po raz pierwszy przedstawiona w pracach B. Moore [89], wykorzystuje macierze gramianów sterowalności i obserwowalności do wyodrębnienia z modelu części dominującej i nieistotnej. Część dominująca związana jest z dużymi wartościami własnymi gramianu sterowalności (stany, których osiągnięcie nie wymaga dużej energii sygnału wejściowego) oraz obserwowalności (stany generujące dużą energię sygnału wyjściowego) [90] [91]. Pozostała część modelu pierwotnego związana ze stanami niesterowalnymi oraz nieobserwowalnymi lub słabo sterowalnymi i słabo obserwowalnymi może zostać pominięta w modelu zredukowanym bez utraty istotnych cech modelu [92]. Wyodrębnienie w macierzach stanu modelu części dominującej i nieistotnej wykonywane jest poprzez wyznaczenie niesobliwych macierzy transformacji przestrzeni stanu \mathbf{T} i \mathbf{L} , które pozwalają uzyskać reprezentację modelu dla zmienionego układu współrzędnych (ang. *similarity transformation*), bez wpływu na charakterystykę wejściowo-wyjściową [92]. Wynikiem zastosowania transformacji zmiennych stanu dla metody BTA jest nowa reprezentacja modelu [93]: $\bar{A} = TAL$, $\bar{B} = TB$, $\bar{C} = CL$, $\bar{D} = D$, w której dla równań macierzowych w przestrzeni stanu można dokonać podziału macierzy stanu na część dominującą i część nieistotną [93] [94].

$$\begin{aligned} \begin{bmatrix} \dot{\bar{x}}_1 \\ \dot{\bar{x}}_2 \end{bmatrix} &= \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} + \begin{bmatrix} \bar{B}_1 \\ \bar{B}_2 \end{bmatrix} u \\ y &= \begin{bmatrix} \bar{C}_1 & \bar{C}_2 \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} + D \end{aligned} \quad (9)$$

Macierze transformacji \mathbf{T} i \mathbf{L} wyznaczone są poprzez równoważenie gramianów sterowalności \mathbf{P} i obserwowalności \mathbf{Q} . W wyniku operacji równoważenia gramiany stają się równymi sobie macierzami diagonalnymi, z wartościami szczególnymi Hankela na przekątnej głównej [95]:

$$\begin{aligned} \bar{P} &= TPT^*, & \bar{Q} &= L^*QL \\ \bar{P} &= \bar{Q} = \text{diag}(\sigma_1, \dots, \sigma_n) \end{aligned} \quad (10)$$

Taka reprezentacja przestrzeni stanu jest nazywana realizacją zrównoważoną (ang. *balanced realization*). Oznacza to, że zrównoważeniu uległa relacja pomiędzy przejściami wejście-osiągnięcie stanu oraz stan-wyjście [95].

Wartości szczególne Hankela na przekątnej zrównoważonych gramianów posortowane są w porządku malejącym $\sigma_i \geq \sigma_{i+1}$. Za model zredukowany można więc przyjąć część modelu związanego z dużymi wartościami szczególnymi Hankela: $A_r = \bar{A}_{11}$, $B_r = \bar{B}_1$, $C_r = \bar{C}_1$, $D_r = D$. Usunięta natomiast zostaje część modelu związana z niewielkimi wartościami szczególnymi Hankela.

Metoda BTA umożliwia oszacowanie maksymalnego błędu aproksymacji, który jest równy podwojonej sumie usuniętych wartości szczególnych Hankela w wyniku procesu redukcji: [20]

$$\|G_o(s) - G_r(s)\|_\infty = 2 \sum_{k=q+1}^N \sigma_k \quad (11)$$

gdzie: k oznacza rząd modelu zredukowanego, natomiast N rząd modelu oryginalnego.

2.3. Algorytmy wyznaczania macierzy transformacji

Ze względu na fakt, że wyznaczanie macierzy transformacji nie jest operacją jednoznaczną, istnieje szereg metod realizujących to zadanie. Algorytmy wyznaczające macierze transformacji mogą zostać podzielone na dwie kategorie [94]:

- Algorytmy wykorzystujące dekompozycję Cholesky'ego gramianów sterowalności lub/i obserwowalności,
- Algorytmy wykorzystujące substytut dekompozycji Cholesky'ego gramianów sterowalności i obserwowalności (dekompozycja gramianów na wektory i wartości własne).

W treści pracy przyjęto nazewnictwo algorytmów zaproponowane w pracy M. Rydla [15].

Metoda RPR została opisana między innymi w [20] [15]. W celu wyznaczenia macierzy transformacji T wykorzystuje dekompozycję Cholesy'ego gramianu obserwowalności. Macierz transformacji L wyznaczana jest jako odwrotność macierzy T . Metoda SR została opisana między innymi w [13] [96] [15]. Zaletą metody w porównaniu z algorytmem RPR jest brak konieczności wyznaczania odwrotności macierzy T , co może być źle uwarunkowane numerycznie w przypadku macierzy wysokiego rzędu [15]. Metody BFSR zostały opisane między innymi w [13] [97] [98] [99] [100]. Główną ich zaletą w stosunku do wcześniej prezentowanej metody SR jest lepsze uwarunkowanie numeryczne. Wynika to z wykonania dekompozycji QR w stosunku do rezultatów uzyskiwanych metodą SR. Wyznaczanie macierzy transformacji równoważącej gramiany dla modeli nie będących realizacją minimalną może okazać się kłopotliwe ze względów numerycznych. W metodzie Schur'a [101] zaproponowano, aby wyznaczyć macierze transformacji bez wcześniejszego równoważenia gramianów. Wykonywane jest to poprzez projekcję modelu na lewą i prawą przestrzeń własną związaną z dużymi wartościami własnymi iloczynu gramianu sterowalności i obserwowalności [101]. Realizacja metody związana jest z koniecznością wykonania wielu ortogonalnych rotacji Givensa [102], co powoduje, że algorytm jest długotrwały. Opisane do tego miejsca metody wymagały wyznaczenia dekompozycji Cholesky'ego jednego lub obu gramianów.

Dekompozycja Cholesky'ego może zostać wyznaczona wyłącznie dla symetrycznych, rzeczywistych, dodatnio określonych macierzy [103]. Gramiany z definicji spełniają ten warunek [104], jednak na skutek błędów numerycznych algorytmów wyznaczających rozwiązanie równania Lapunowa, może okazać się, że niektóre wartości własne wyznaczonego gramianu posiadają bliskie zeru wartości ujemne. Wyznaczenie dekompozycji Cholesky'ego takiej macierzy nie powiedzie się. W książce [105] Obinata i Anderson zaproponowali podejście, aby zamiast wykorzystywać dekompozycję Cholesky'ego gramianów, zastosować jej substytut wyznaczony na bazie wektorów i wartości własnych, również pozwalających zapisać gramian w postaci iloczynu macierzy $\mathbf{X}=\mathbf{Y}^*\mathbf{Y}$. Dekompozycja gramianów na wektory i wartości własne może zostać zapisana jako [105]:

$$\mathbf{P}=\mathbf{V}_c \mathbf{D}_c \mathbf{V}_c^*, \quad \mathbf{Q}=\mathbf{V}_o \mathbf{D}_o \mathbf{V}_o^* \quad (12)$$

gdzie: $\mathbf{V}_c, \mathbf{V}_o$ – macierze prawych wektorów własnych, $\mathbf{D}_c, \mathbf{D}_o$ – diagonalne macierze wartości własnych. Równania (12) można zapisać w postaci:

$$\mathbf{P}=\mathbf{V}_c \mathbf{D}_c \mathbf{V}_c^* = \left(\mathbf{V}_c \mathbf{D}_c^{1/2}\right)\left(\mathbf{V}_c \mathbf{D}_c^{1/2}\right)^* = \tilde{\mathbf{S}}^* \tilde{\mathbf{S}}, \quad \mathbf{Q}=\mathbf{V}_o \mathbf{D}_o \mathbf{V}_o^* = \left(\mathbf{V}_o \mathbf{D}_o^{1/2}\right)\left(\mathbf{V}_o \mathbf{D}_o^{1/2}\right)^* = \tilde{\mathbf{R}}^* \tilde{\mathbf{R}} \quad (13)$$

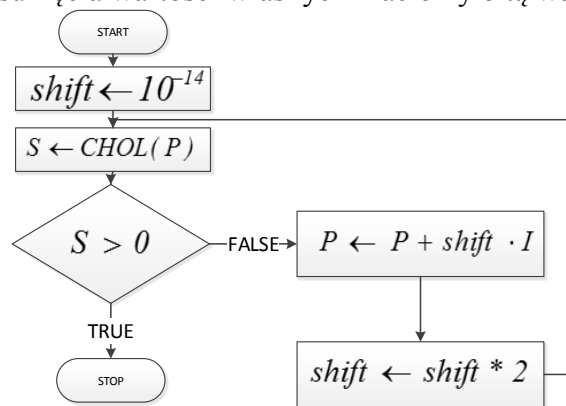
przy czym \tilde{S}^* i \tilde{R}^* stanowią substytut dekompozycji Cholesky'ego dla gramianów.

Analogicznie jak dla algorytmu Obinata-Andersona, metody EIG-SR, EIG-BFSR wykorzystują substytut dekompozycji Cholesky'ego gramianów w miejsce klasycznej dekompozycji Cholesky'ego w odniesieniu do algorytmów SR i BFSR.

2.3.1. Metoda wyznaczania dekompozycji Cholesky'ego z nieznacznym przesunięciem wartości własnych

Dla klasycznych metod redukcji: RPR, SR oraz BFSR stwierdzono problemy numeryczne związane z wyznaczaniem dekompozycji Cholesky'ego gramianów przy rzędzie modelu pierwotnego powyżej 150 zmiennych stanu. Wyklucza to zastosowanie oryginalnych algorytmów. Zauważono, że powodem przez który algorytmy nie działają właściwie są ujemne wartości własne o niewielkich wartościach (powyżej -10^{-10}).

Autor zaproponował metodę, która umożliwia doprowadzenie wyznaczonych gramianów do postaci macierzy dodatnio określonych, poprzez zastosowanie niewielkich przesunięć wartości własnych. Bazuje ona na fakcie, że gramiany są macierzami symetrycznymi, w związku z czym dodanie takiej samej wartości do wszystkich elementów diagonalni prowadzi do przesunięcia wartości własnych macierzy o tą wartość (rys. 11) [106].



RYSUNEK 11. SCHEMAT BLOKOWY ALGORYTMU DOPROWADZAJĄCEGO GRAMIAN DO POSTACI MACIERZY DODATNIO OKREŚLONEJ

Ze względu na niewielką wartość przesunięcia (rzędu 10^{-12} - 10^{-10}) wpływ operacji na właściwości modelu może zostać pominięty. Wynik działania algorytmu przekazywany jest metodom wyznaczającym macierze transformacji jako dekompozycja Cholesky'ego gramianów.

2.4. Redukcja modeli z wykorzystaniem metody częstotliwościowych funkcji wagowych (ang. Frequency Weighted)

Metoda FW (ang. Frequency Weighted) została opracowana w celu redukcji rzędu modeli przy minimalizacji błędu aproksymacji wyrażonego zależnością [13]:

$$\|G_{WO}(s)(G_o(s) - G_r(s))G_{WI}(s)\|_{\infty} \quad (14)$$

Gdzie: $G_o(s)$ – transmitancja modelu oryginalnego, $G_r(s)$ – transmitancja modelu zredukowanego, $G_{WO}(s)$ – transmitancja filtru wyjściowego, $G_{WI}(s)$ – transmitancja filtru wejściowego.

Obecność filtrów wejściowych i wyjściowych powoduje, że model zredukowany będzie dokładniej aproksymował charakterystyki częstotliwościowe, dla których filtr wyjściowy lub/i filtr wejściowy posiadają większe wartości szczególne [13].

Metoda została po raz pierwszy przedstawiona w pracach D. Ennsa [107]. Dla filtrów wejściowego i wyjściowego, których reprezentacje w postaci równań w przestrzeni stanu są określone zależnościami [108]:

$$W_i : \begin{cases} \dot{x}_i(t) = A_i x_i(t) + B_i u_i(t) \\ y_i(t) = C_i x_i(t) + D_i u_i(t) \end{cases} \quad W_o : \begin{cases} \dot{x}_o(t) = A_o x_o(t) + B_o u_o(t) \\ y_o(t) = C_o x_o(t) + D_o u_o(t) \end{cases} \quad (15)$$

reprezentacja systemu pierwotnego z dołączonymi filtrami może zostać zapisana jako [108]:

$$W_o \Sigma W_i : \begin{cases} \dot{\bar{x}}(t) = \bar{A} \bar{x}(t) + \bar{B} \bar{u}(t) \\ \bar{y}(t) = \bar{C} \bar{x}(t) + \bar{D} \bar{u}(t) \end{cases} \quad (16)$$

gdzie [109]:

$$\bar{A} = \begin{bmatrix} A_o & B_o C & B_o D C_i \\ 0 & A & B C_i \\ 0 & 0 & A_i \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} B_o D D_i \\ B D_i \\ B_i \end{bmatrix}, \quad \bar{C} = [C_o \quad D_o C \quad D_o D C_i], \quad \bar{D} = D_o D D_i, \quad (17)$$

W algorytmie Ennsa wyznaczenie gramianów dla systemu (16) odbywa się z wykorzystaniem gramianów wyznaczonych z równań Lapunowa [110]:

$$\bar{A} \bar{P} + \bar{P} \bar{A}^* + \bar{B} \bar{B}^* = 0, \quad \bar{A}^* \bar{Q} + \bar{Q} \bar{A} + \bar{C}^* \bar{C} = 0 \quad (18)$$

Obliczone gramiany mają postać określoną zależnościami [109]:

$$\bar{P} = \begin{bmatrix} P_o & P_{12} & P_{13} \\ P_{12}^T & P & P_{23} \\ P_{13}^T & P_{23}^T & P_i \end{bmatrix}, \quad \bar{Q} = \begin{bmatrix} Q_o & Q_{12} & Q_{13} \\ Q_{12}^T & Q & Q_{23} \\ Q_{13}^T & Q_{23}^T & Q_i \end{bmatrix} \quad (19)$$

Muszą one zostać doprowadzone do rzędu modelu zredukowanego, co jest realizowane za pomocą zależności [110]:

$$P = \begin{bmatrix} 0_{n \times n_o} & I_{n \times n} & 0_{n \times n_i} \end{bmatrix} \bar{P} \begin{bmatrix} 0_{n_o \times n} \\ I_{n \times n} \\ 0_{n_i \times n} \end{bmatrix}, \quad Q = \begin{bmatrix} 0_{n \times n_o} & I_{n \times n} & 0_{n \times n_i} \end{bmatrix} \bar{Q} \begin{bmatrix} 0_{n_o \times n} \\ I_{n \times n} \\ 0_{n_i \times n} \end{bmatrix} \quad (20)$$

W dalszej kolejności przy pomocy opisanych w podrozdziale 2.3 metod wyznaczane są macierze T i L równoważące model Σ .

Możliwe jest też wyznaczanie gramianów na podstawie modelu systemu pierwotnego z dołączonym filtrem wejściowym dla gramianu sterowalności i wyjściowym dla gramianu obserwowalności. W tym przypadku istnieje konieczność wyznaczenia dwóch reprezentacji modelu pierwotnego z dołączonym filtrem wejściowym [13]:

$$\Sigma W_i : \begin{cases} \dot{\bar{x}}_i(t) = \bar{A}_i \bar{x}_i(t) + \bar{B}_i \bar{u}_i(t) \\ \bar{y}_i(t) = \bar{C}_i \bar{x}_i(t) + \bar{D}_i \bar{u}_i(t) \end{cases} \quad (21)$$

gdzie [13]:

$$\bar{A}_i = \begin{bmatrix} A & B C_i \\ 0 & A_i \end{bmatrix}, \quad \bar{B}_i = \begin{bmatrix} B D_i \\ B_i \end{bmatrix}, \quad \bar{C}_i = [C \quad D C_i], \quad \bar{D}_i = D D_i, \quad (22)$$

oraz wyjściowym [13]:

$$W_o \Sigma : \begin{cases} \dot{\bar{x}}_o(t) = \bar{A}_o \bar{x}_o(t) + \bar{B}_o \bar{u}_o(t) \\ \bar{y}_o(t) = \bar{C}_o \bar{x}_o(t) + \bar{D}_o \bar{u}_o(t) \end{cases} \quad (23)$$

gdzie [13]:

$$\bar{A}_o = \begin{bmatrix} A & 0 \\ B C_o & A_o \end{bmatrix}, \quad \bar{B}_o = \begin{bmatrix} B \\ B_o D \end{bmatrix}, \quad \bar{C}_o = [D_o C \quad C_o], \quad \bar{D}_o = D_o D, \quad (24)$$

W tym przypadku gramiany dla modelu z filtrem wyznaczane są z równań Lapunowa [111]:

$$\bar{A}_i \bar{P}_i + \bar{P}_i \bar{A}_i^* + \bar{B}_i \bar{B}_i^* = 0 \quad (25)$$

$$\bar{A}_o^* \bar{Q}_o + \bar{Q}_o \bar{A}_o + \bar{C}_o^* \bar{C}_o = 0 \quad (26)$$

Obliczone gramiany mają postać określoną zależnościami [111]:

$$\bar{P}_i = \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^* & P_i \end{bmatrix}, \quad \bar{Q}_o = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^* & Q_o \end{bmatrix} \quad (27)$$

Dla metod wyznaczających macierze transformacji przekazywane są podmacierze P_{11} jako gramian sterowalności oraz Q_{11} jako gramian obserwowalności.

Metody FW przedstawione dotychczas gwarantują stabilność modelu zredukowanego przy wykorzystaniu tylko jednego filtra (albo wejściowego, albo wyjściowego) [112].

Modyfikacja metody Ennsa, która pozwala na zachowanie stabilności modelu zredukowanego przy wykorzystaniu obu filtrów została zaproponowana w pracy [22]. Bazuje ona na wyznaczaniu gramianów sterowalności z zależności [113]:

$$A(P_{11} - P_{12}P_i^{-1}P_{12}^*) + (P_{11} - P_{12}P_i^{-1}P_{12}^*)A^* + P_x P_x^* = 0 \quad (28)$$

gdzie:

$$P_x = BD_i - P_{12}P_i^{-1}B_i \quad (29)$$

oraz gramianu obserwowalności z zależności [113]:

$$A^*(Q_{11} - Q_{12}^*Q_o^{-1}Q_{12}) + (Q_{11} - Q_{12}^*Q_o^{-1}Q_{12})A + Q_Y^*Q_Y = 0 \quad (30)$$

gdzie:

$$Q_Y = D_o C - C_o Q_o^{-1}Q_{12} \quad (31)$$

W pracy [13] została przedstawiona metoda, wprowadzająca dodatkowe skalowanie wartości macierzy $P_{12}P_i^{-1}P_{12}^*$ oraz $Q_{12}^*Q_o^{-1}Q_{12}$. Gramiany sterowalności i obserwowalności wyznaczane są z zależności:

$$A(P_{11} - \alpha P_{12}P_i^{-1}P_{12}^*) + (P_{11} - \alpha P_{12}P_i^{-1}P_{12}^*)A^* + P_x P_x^* = 0 \quad (32)$$

$$A^*(Q_{11} - \beta Q_{12}^*Q_o^{-1}Q_{12}) + (Q_{11} - \beta Q_{12}^*Q_o^{-1}Q_{12})A + Q_Y^*Q_Y = 0 \quad (33)$$

Przy czym współczynniki α i β określają stopień wpływu macierzy $P_{12}P_i^{-1}P_{12}^*$ oraz $Q_{12}^*Q_o^{-1}Q_{12}$ na wynik rozwiązania równania Lapunowa. Dla wartości $\alpha = \beta = 0$ równanie odpowiada algorytmowi [107] natomiast dla $\alpha = \beta = 1$ równanie odpowiada algorytmowi [22].

Ostatnia z prezentowanych metod wprowadzona w [113] również gwarantuje stabilność modelu zredukowanego przy redukcji z użyciem obu filtrów. Bazuje ona na rozwinięciu równań Lapunowa (25) (26) do postaci [113]:

$$AP_{11} + P_{11}A^* + BC_i P_{12} + P_{12}^* C_i^* B^* + BD_i D_i^* B^* = 0 \quad (34)$$

$$A^* Q_{11} + Q_{11}A + Q_{12} B_o C + C^* B_o^* Q_{12}^* + C^* D_o^* D_o C = 0 \quad (35)$$

Przyjmując [113]:

$$X = BC_i P_{12} + P_{12}^* C_i^* B^* + BD_i D_i^* B^* \quad (36)$$

$$Y = Q_{12} B_o C + C^* B_o^* Q_{12}^* + C^* D_o^* D_o C \quad (37)$$

można wykonać dekompozycję SVD symetrycznych macierzy X i Y [113]:

$$X = USU^* \quad (38)$$

$$Y = VHV^* \quad (39)$$

w celu wyznaczenia :

$$\tilde{B} = U \begin{bmatrix} \sqrt{|s_1|} & 0 & \dots & 0 \\ 0 & \ddots & & \\ & & \sqrt{|s_M|} & \ddots & \vdots \\ \vdots & & \ddots & 0 & \\ 0 & \dots & & \ddots & 0 & 0 \end{bmatrix}, \quad \tilde{C} = \begin{bmatrix} \sqrt{|h_1|} & 0 & \dots & 0 \\ 0 & \ddots & & \\ & & \sqrt{|h_N|} & \ddots & \vdots \\ \vdots & & \ddots & 0 & \\ 0 & \dots & & \ddots & 0 & 0 \end{bmatrix} V^* \quad (40)$$

gdzie:

$s_1 \dots s_M$ - wartości szczególne zawarte w macierzy S ,

$h_1 \dots h_N$ - wartości szczególne zawarte w macierzy H ,

M – rząd macierzy X , N – rząd macierzy Y .

W kolejnym kroku następuje rozwiązanie równania Lapunowa:

$$A\hat{P} + \hat{P}A^* + \tilde{B}\tilde{B}^* = 0, \quad \hat{Q}A + A^*\hat{Q} + \tilde{C}^*\tilde{C} = 0 \quad (41)$$

Na podstawie gramianów \hat{P} oraz \hat{Q} wyznaczane są macierze transformacji T i L w celu zrównoważenia modelu.

2.5. Uwagi

Metody redukcji modeli LTI oparte na SVD, znane od 1981 roku [89], zapewniają uzyskanie stabilnego modelu zredukowanego dla stabilnego modelu pierwotnego. Tej właściwości nie posiadają inne metody redukcji. Ponadto metody redukcji oparte na SVD dają możliwość oszacowania maksymalnego błędu aproksymacji, równego podwojonej sumie usuniętych w wyniku redukcji wartości szczególnych Hankela. Metody redukcji oparte na SVD bazują na wyznaczeniu gramianów sterowalności i obserwowalności dla modelu, a następnie zrównoważeniu gramianów poprzez zastosowanie przekształcenia wektora stanu, aby obydwa gramiany zostały doprowadzone do postaci diagonalnej. Niestety operacje wyznaczania gramianów charakteryzują się znaczną złożonością obliczeniową ($O(n^3)$), co pociąga za sobą długie czasy obliczeń. Dzięki zastosowaniu różnych metod dekompozycji modeli i zastosowaniu obliczeń równoległych można znacznie skrócić czas obliczeń. Daje to możliwość zastosowania redukcji modeli opartej na SVD dla modeli LTI wysokiego rzędu.

Dla modeli podsystemów parownika kotła energetycznego BP-1150 stwierdzono problemy numeryczne związane z realizacją klasycznych metod (RPR, SR, BFSR) wyznaczania macierzy transformacji T i L w celu redukcji. Powodem wystąpienia problemów jest błędna interpretacja macierzy gramianów jako macierzy niedodatnio określonych, dla których wyznaczenie dekompozycji Cholesky'ego jest niemożliwe. Zaproponowane w pracy podejście pozwala, poprzez modyfikację elementów znajdujących się na przekątnej macierzy gramianów, uzyskać macierze dodatnio określone, dla których istnieje dekompozycja Cholesky'ego. Wprowadzony w wyniku przeprowadzonych operacji błąd jest na tyle niewielki, że wpływ przekształcenia na właściwości modelu może zostać pominięty.

Bezpośrednia redukcja niestabilnych modeli z wykorzystaniem opisanych metod SVD jest niemożliwa. Podejście, które pozwala na wykonanie redukcji modelu niestabilnego związane jest z dekompozycją modelu na część stabilną i niestabilną, redukcji części stabilnej i scaleniu części zredukowanej z podsystemem niestabilnym [114].

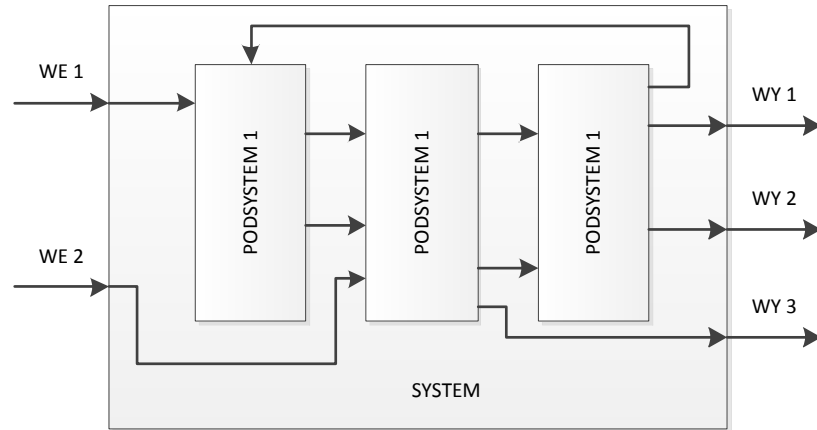
Metoda FW pozwala na uzyskanie modeli zredukowanych dokładnie aproksymujących charakterystyki częstotliwościowe modelu pierwotnego w określonym przedziale częstotliwości. Jak wykazano w pracy [15] błąd aproksymacji dla modeli uzyskanych metodą FW w znacznym stopniu zależy od parametrów częstotliwościowych

funkcji wagowych (filtrów). Niestety problem wyznaczenia optymalnych parametrów częstotliwościowych funkcji wagowych nie jest problemem wypukłym (ang. non-convex problem) [115], co oznacza, że nie istnieje jednoznaczne rozwiązanie. Problem ten został rozwiązany w pracy [15] dzięki zastosowaniu algorytmów ewolucyjnych. Wyznaczenie optymalnych parametrów częstotliwościowych funkcji wagowych wymaga wielokrotnych operacji redukcji modelu z zastosowaniem algorytmu ewolucyjnego, co znacznie wydłuża czas obliczeń.

3. Metody jednopoziomowej dekompozycji modeli LTI

3.1. Dekompozycja fizyczna (ang. interconnected systems)

Dekompozycja fizyczna modelu obiektu złożonego oparta jest na wydzieleniu podsystemów, dla których oddzielnie zostają wyznaczone modele w przestrzeni stanów (ang. *interconnected systems*). Model obiektu złożonego uzyskuje się poprzez łączenie odpowiednich wyjść podsystemów z wejściami pozostałych podsystemów (rys. 12).



RYSUNEK 12. PRZYKŁAD DEKOMPOZYCJI FIZYCZNEJ

Dekompozycja fizyczna modelu parownika kotła energetycznego BP-1150 została opisana w rozdziale 6.

3.2. Dekompozycja strukturalna (ang. coupled systems)

W celu zmniejszenia złożoności obliczeniowej związanej z redukcją modeli podsystemów, uzyskanych w wyniku dekompozycji fizycznej, można dokonać dekompozycji strukturalnej modelu, polegającej na podziale systemu na mniejsze części, gdzie rozmiar poszczególnych podsystemów jest wielokrotnością rozmiaru elementu skończonego. Dekompozycja strukturalna bazuje na podziale macierzy modelu oryginalnego, opisanego za pomocą równań w przestrzeni stanu, na większą liczbę modeli połączonych ze sobą interfejsami, wynikającymi ze struktury macierzy A modelu oryginalnego.

Dla modelu oryginalnego określonego za pomocą równań macierzowych:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \quad (42)$$

$$y = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (43)$$

gdzie:

$$\Gamma_{2,1} = A_{12}, \quad \Gamma_{1,2} = A_{21} \quad (44)$$

można wyodrębnić dwa podsystemy:

$$S_1 : \begin{cases} \dot{x}_1 = A_{11}x_1 + B_1 u_1 + \Gamma_{2,1} x_2 \\ y_1 = C_1 x_1 \end{cases}, \quad S_2 : \begin{cases} \dot{x}_2 = A_{22}x_2 + B_2 u_2 + \Gamma_{1,2} x_1 \\ y_2 = C_2 x_2 \end{cases} \quad (45)$$

Redukcja systemu oryginalnego może zostać sprowadzona do przeprowadzenia niezależnych procesów redukcji dwóch wydzielonych modeli S_1 oraz S_2 , po czym połączenia

zredukowanych modeli w jeden model, z uwzględnieniem zmodyfikowanych interfejsów. Podejście to może zostać zrealizowane na jeden z dwóch sposobów:

- niezależną redukcję obu podsystemów bez uwzględniania interfejsów oraz przekształcenie macierzy interfejsów zgodnie z wyznaczonymi macierzami transformacji (dekompozycja strukturalna uproszczona – *StrUpr*),
- zmodyfikowanie macierzy wejść oraz wyjść modeli, aby uwzględniały wzajemne powiązania podsystemów, przeprowadzenie procesu redukcji oraz wyznaczenie interfejsów z wykorzystaniem wyznaczonych macierzy transformacji (dekompozycja strukturalna – *Str*).

3.2.1. Dekompozycja strukturalna uproszczona (*StrUpr*)

Ten sposób dekompozycji może zostać zastosowany do redukcji modeli, w których wydzielone części są zarówno sterowalne jak i obserwowalne. Redukcji podlegają obydwie podsystemy, bez uwzględnienia interfejsów. Uzyskane zostają następujące modele zredukowane:

$$\tilde{S}_1 : \begin{cases} \tilde{x}_1 = \tilde{A}_{11} \tilde{x}_1 + \tilde{B}_1 u \\ \tilde{y}_1 = \tilde{C}_1 \tilde{x}_1 \end{cases}, \quad \tilde{S}_2 : \begin{cases} \tilde{x}_2 = \tilde{A}_{22} \tilde{x}_2 + \tilde{B}_2 u \\ \tilde{y}_2 = \tilde{C}_2 \tilde{x}_2 \end{cases} \quad (46)$$

gdzie: $\tilde{x}_1 = T_1 x_1, \quad x_1 = L_1 \tilde{x}_1, \quad \tilde{x}_2 = T_2 x_2, \quad x_2 = L_2 \tilde{x}_2, \quad (47)$

$$\tilde{A}_{11} = T_1 A_{11} L_1, \quad \tilde{B}_1 = T_1 B_1, \quad \tilde{C}_1 = C_1 L_1, \quad (48)$$

$$\tilde{A}_{22} = T_2 A_{22} L_2, \quad \tilde{B}_2 = T_2 B_2, \quad \tilde{C}_2 = C_2 L_2. \quad (49)$$

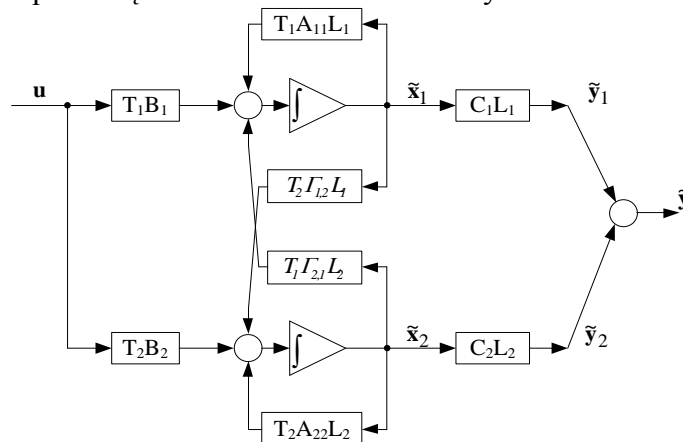
Macierze T_1 , L_1 oraz T_2 , L_2 , wyznaczone w wyniku redukcji obydwu podsystemów, wykorzystywane są do przekształcenia macierzy interfejsów zgodnie z zależnościami:

$$\tilde{F}_{2,1} = T_1 F_{2,1} L_2, \quad \tilde{F}_{1,2} = T_2 F_{1,2} L_1 \quad (50)$$

Zredukowane modele podsystemów, z uwzględnieniem wzajemnych powiązań, przyjmują postać wyrażoną następująco:

$$\tilde{S}_1 : \begin{cases} \tilde{x}_1 = \tilde{A}_{11} \tilde{x}_1 + \tilde{B}_1 u + \tilde{F}_{2,1} \tilde{x}_2 \\ \tilde{y}_1 = \tilde{C}_1 \tilde{x}_1 \end{cases}, \quad \tilde{S}_2 : \begin{cases} \tilde{x}_2 = \tilde{A}_{22} \tilde{x}_2 + \tilde{B}_2 u + \tilde{F}_{1,2} \tilde{x}_1 \\ \tilde{y}_2 = \tilde{C}_2 \tilde{x}_2 \end{cases} \quad (51)$$

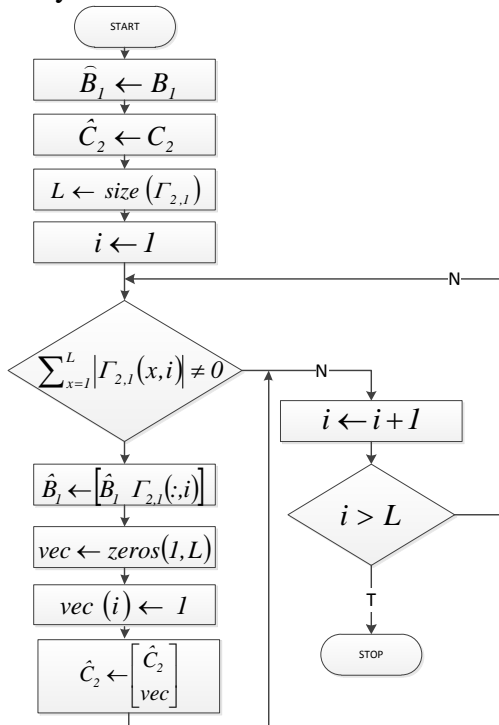
Rysunek 13 obrazuje sposób łączenia modeli zredukowanych.



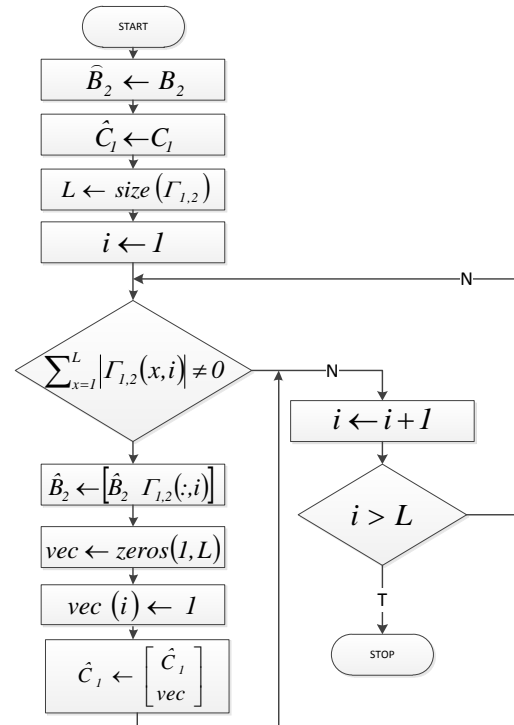
RYСУNEK 13. SPOSÓB ŁĄCZENIA MODELI ZREDUKOWANYCH DLA DEKOMPOZYCJI STRUKTURALNEJ UPROSzcZONEJ

3.2.2. Dekompozycja strukturalna (Str)

W przypadku, gdy dekompozycja strukturalna wydzieliła dwa modele, z których choćby jeden jest niesterowalny/nieobserwowalny konieczne jest zmodyfikowanie macierzy wejść i wyjść modeli. Odbyna się to poprzez wyznaczenie kolumn w $\Gamma_{2,1}$ oraz $\Gamma_{1,2}$, w których przynajmniej jeden element jest różny od zera. Indeksy wyznaczonych kolumn określają numery współrzędnych wektora stanu \mathbf{x}_2 oraz \mathbf{x}_1 które należy wprowadzić do wektorów wejściowych modeli poprzez modyfikację macierzy \mathbf{B}_1 i \mathbf{B}_2 . Indeksy te wykorzystywane są również do modyfikacji struktury macierzy wyjść \mathbf{C}_1 i \mathbf{C}_2 . Algorytm modyfikujący strukturę macierzy wyjść i wejść obydwu podsystemów został przedstawiony na rysunku 14 i 15.



RYSUNEK 14. MODYFIKACJA MACIERZY B_1 I C_2

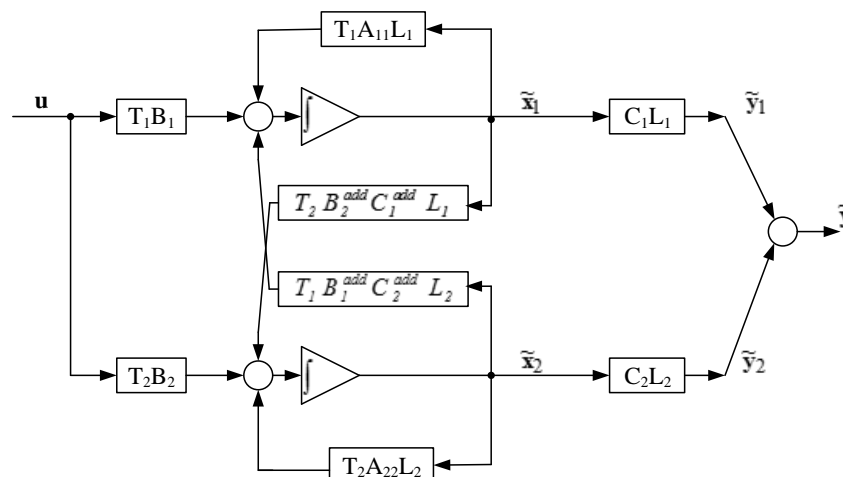


RYSUNEK 15. MODYFIKACJA MACIERZY B_2 I C_1

W kolejnym kroku następuje redukcja obydwu modeli ze zmodyfikowanymi macierzami wejść i wyjść:

$$\hat{S}_1 : \begin{cases} \dot{x}_1 = A_{11} x_1 + \hat{B}_1 u \\ y_1 = \hat{C}_1 x_1 \end{cases}, \quad \hat{S}_2 : \begin{cases} \dot{x}_2 = A_{22} x_2 + \hat{B}_2 u \\ y_2 = \hat{C}_2 x_2 \end{cases} \quad (52)$$

Jeżeli kolumny dodane do macierzy B_1 i B_2 zostaną oznaczone odpowiednio poprzez B_1^{add} oraz B_2^{add} , natomiast wiersze dodane do macierzy C_1 oraz C_2 przez C_1^{add} oraz C_2^{add} , połączenie zredukowanych modeli odbywa się zgodnie ze schematem zobrazowanym na rysunku 16.



RYСУNEK 16. ŁĄCZENIE MODELI ZREDUKOWANYCH Z UWZGLĘDNIENIEM ZMODYFIKOWANYCH MACIERZY B I C

3.3. Dekompozycja modeli LTI ze względu na wartości własne (ang. slow-fast)

Modele systemów LTI mogą opisywać zarówno szybkie procesy dynamiczne jak i procesy wolnozmiennie, przebiegające równocześnie w modelowanym obiekcie. W przypadku, gdy w rozważanym modelu występują zarówno mody szybkie jak i wolne, określany jest on mianem modelu sztywnego. Współczynnik sztywności modelu może zostać wyznaczony z zależności [116]:

$$\eta(A) = \frac{\text{Max}\{\text{Re}(\lambda_i)\}}{\text{Min}\{\text{Re}(\lambda_i) \neq 0\}} \quad (53)$$

Określa on stosunek wartości bezwzględnej części rzeczywistej największej wartości własnej macierzy stanu A do najmniejszej. W rozważanym modelu parownika kotła energetycznego BP-1150 występują szybkie procesy oscylacyjne.

Modele cechujące się znaczną sztywnością są trudne do analizowania metodami numerycznymi. Rozwiązanie tego problemu może polegać na dekompozycji systemu na dwie części – jedną związaną z procesami szybkimi i drugą zawierającą opis procesów wolnych. [117] [118] [119] W literaturze szeroko opisany jest sposób dekompozycji modelu, w którego strukturze wydzielono część szybką i część wolną. Struktura takiego modelu może zostać przedstawiona za pomocą równań:

$$\begin{bmatrix} \dot{\mathbf{x}}_S \\ \varepsilon \dot{\mathbf{x}}_F \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{SS} & \mathbf{A}_{SF} \\ \mathbf{A}_{FS} & \mathbf{A}_{FF} \end{bmatrix} \begin{bmatrix} \mathbf{x}_S \\ \mathbf{x}_F \end{bmatrix} + \begin{bmatrix} \mathbf{B}_S \\ \mathbf{B}_F \end{bmatrix} \mathbf{u} \quad (54)$$

$$y = \begin{bmatrix} C_S & C_F \end{bmatrix} \begin{bmatrix} x_S \\ x_F \end{bmatrix}, \quad (55)$$

gdzie równania związane z częścią wektora stanu $x_S \in R^{n_1}$ związane są z procesami wolnymi, natomiast równania związane z częścią wektora stanu $x_F \in R^{n_2}$ związane są z procesami szybkimi. Wartość ε ($0 < \varepsilon \ll 1$) określa stopień odizolowania procesów szybkich i wolnych.

W literaturze opisującej problematykę dekompozycji slow-fast autorzy zakładają, że równania opisujące procesy szybkie i wolne są rozdzielone w strukturze modelu. Posługując się metodą elementów skończonych do wyznaczenia modelu obiektu fizycznego, w ramach równań opisujących jeden element skończony znajdują się szybko- i wolnozmiennie zmienne

stanu. Koniecznym staje się przenieś numerowanie zmiennych stanu w taki sposób, aby równania opisujące szybką dynamikę znalazły się w jednej części modelu, natomiast równania związane z procesami wolnymi w drugiej części. W pracy autor zaproponował zastosowanie dekompozycji Epsilon do wydzielenia części wolnej i szybkiej modelu.

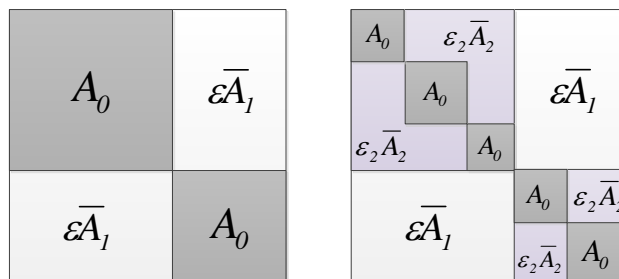
3.4. Dekompozycja Epsilon

Dekompozycja Epsilon została szeroko opisana w pracach Dragoslava Šiljaka [30] [120] [121] [122]. Autor wykorzystuje fakt, że złożone modele obiektów LTI cechują się często słabym powiązaniem (lub brakiem powiązania) dla większości zmiennych stanu. Zazwyczaj dynamika obiektu jest w znacznym stopniu zdominowana przez silnie powiązane ze sobą zmienne stanu. Kluczowe staje się więc wyznaczenie grup silnie ze sobą powiązanych elementów. Na tej podstawie wydzielone zostają podsystemy silnie związane wewnętrznymi zależnościami oraz słabo powiązane pomiędzy sobą. Parametrem decydującym o tym, które z powiązań pomiędzy podsystemami uznać za nieistotne jest liczba ε , której wartość mieści się w przedziale od 0 do 1 (wartość ε zostaje arbitralnie ustalona). Wykonywane jest przenieś numerowanie zmiennych stanu w taki sposób, aby elementy silnie ze sobą powiązane (znaczące wartości a_{ij}) znalazły się w blokach leżących na diagonalu w macierzy stanu A (każda grupa silnie powiązanych zmiennych stanu w osobnym bloku). Bloki leżące poza główną przekątną, po wykonaniu operacji permutacji wierszy i kolumn, będą zawierały niewielkie wartości a_{ij} . Dekompozycja epsilon macierzy A może zostać zatem zapisana jako suma blokowo-diagonalnej macierzy A_0 oraz macierzy $\varepsilon\bar{A}_1$ stanowiącej macierz połączeń pomiędzy wydzielonymi blokami.

$$A = A_0 + \varepsilon\bar{A}_1 \quad (56)$$

Wielopoziomowa dekompozycja epsilon polega na wykonaniu analogicznego podziału w odniesieniu do wyodrębnionych podsystemów (rys. 17).

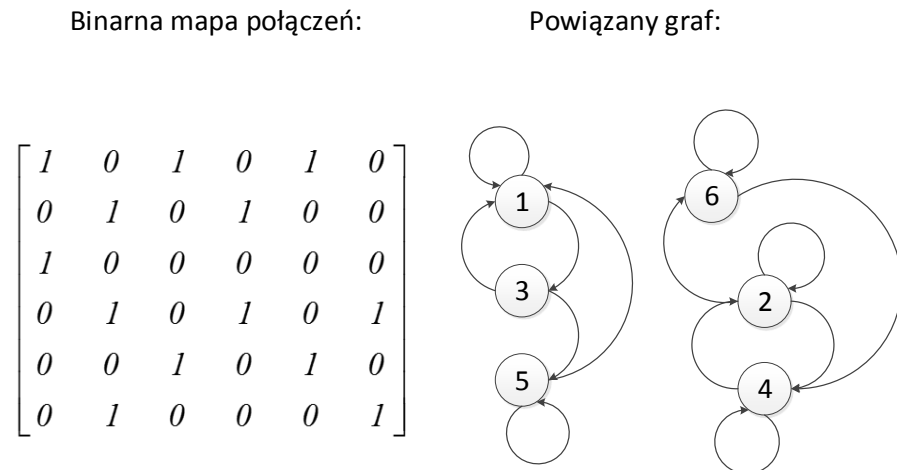
$$A = A_0 + \varepsilon_1\bar{A}_1 + \varepsilon_2\bar{A}_2 + \dots + \varepsilon_n\bar{A}_n \quad (57)$$



RYСУNEK 17. STRUKTURA MACIERZY A PO WYKONANIU JEDNOPOZIOMOWEJ ORAZ DWUPOZIOMOWEJ DEKOMPOZYCJI EPSILON

Pierwszą operacją realizowaną przez algorytm dekompozycji epsilon jest normalizacja wszystkich wierszy macierzy A poprzez dzielenie elementów znajdujących się w wierszu przez element o maksymalnej wartości bezwzględnej w danym wierszu. Po wykonaniu tej czynności wyznaczona zostanie macierz A_{norm} . W kolejnym kroku algorytm zeruje wszystkie elementy macierzy A_{norm} , które są mniejsze od zadanej wartości parametru epsilon. Tworzona jest macierz binarna A_{bin} określająca powiązania zmiennych stanu. W tym celu wszystkim elementom macierzy A_{norm} większym od ε przyporządkowywana jest wartość 1.

Przy użyciu procedury *connected_components* (zaimplementowany algorytm wykorzystuje procedurę z biblioteki BOOST), wyszukiwane są powiązania między wierszami oraz kolumnami macierzy A_{bin} . W tym celu macierz A_{bin} konwertowana jest do postaci grafu, gdzie wierzchołkami grafu są odpowiednio kolumny lub wiersze (dwa wywołania procedury dla macierzy A_{bin} oraz A_{bin}^T). Wartości 1 określają krawędzie grafu – w przypadku gdy w n-tej kolumnie występuje wartość 1 w m-tym wierszu, oznacza to, że istnieje krawędź od n-tego do m-tego wierzchołka grafu. Przykładowa struktura macierzy połączeń i powiązanego z nią grafu została przedstawiona na rysunku 18.



RYSUNEK 18. MACIERZ POŁĄCZEŃ ORAZ WYNIKAJĄCY Z NIEJ GRAF

Procedura szuka w powstałej strukturze podgrafów niepołączonych wzajemnie. Do każdego z wyznaczonych podgrafów zostaje przypisany numer porządkowy. Numer ten zostaje wykorzystany do przypisania indeksu dla każdego wierzchołka w celu określenia jego przynależności do konkretnego podgrafu. W wyniku działania procedury powstają dwa wektory *COL_connect* oraz *ROW_connect*, które określają numery bloków, do których przynależy dany wiersz i kolumna. Przykładowo dla struktury z rysunku 18 wektor *COL_connect* będzie miał następującą postać: $COL_connect = [1 \ 2 \ 1 \ 2 \ 1 \ 2]$

Na podstawie wektorów *COL_connect* oraz *ROW_connect* wyznaczane są wektory permutacji kolumn oraz wierszy, a także rozmiary poszczególnych bloków. Ostatnim krokiem dekompozycji epsilon jest realizacja permutacji macierzy stanu na podstawie wyznaczonych wektorów permutacji.

3.5. Dekompozycja Epsilon slow-fast (*Esf*)

W klasycznej dekompozycji epsilon, w celu wyznaczenia binarnej macierzy połączeń, dla każdego wiersza macierzy A wyznaczany jest element o maksymalnej wartości bezwzględnej, a następnie wszystkie elementy danego wiersza są dzielone przez element maksymalny (standaryzacja). Na skutek tego wszystkie elementy danego wiersza mają wartość bezwzględną mniejszą lub równą 1. Taka sytuacja ma miejsce w każdym wierszu, niezależnie od bezwzględnej wartości elementów. Np. w jednym wierszu poszczególne elementy mają niewielkie wartości bezwzględne, podczas gdy w innym wierszu niektóre elementy mają bardzo duże wartości bezwzględne. W wyniku tak przeprowadzonej operacji standaryzacji niektóre istotne (silne) powiązania między zmiennymi stanu zostają niezauważone.

Jako przykład można przedstawić równania modelu zawierającego elementy skończone w strefie dogrzewu rur ekranowych parownika, reprezentujące bilanse pędu, masy i energii czynnika roboczego (wody), a także temperatury ścianki rury:

$$\frac{d}{dt} \begin{bmatrix} \Delta M_1 \\ \Delta P_1 \\ \Delta h_1 \\ \Delta \Theta_1^j \\ \dots \\ \Delta M_2 \\ \Delta P_2 \\ \Delta h_2 \\ \Delta \Theta_2^j \\ \dots \end{bmatrix} = \begin{bmatrix} -3.72 & 7.93 \cdot 10^4 & 2.47 & 0 & 0 & 0 & \dots & 0 & -7.93 \cdot 10^4 & \dots & 0 & \dots & 0 \\ -5.80 & -0.3 & -6.5 & 13.8 & 0 & 0 & \dots & 0 & 0 & & & & \\ -0.83 & 0.17 & -3.06 & 6.50 & 0 & 0 & \dots & 0 & 0 & & & & \\ -0.008 & 0.10 & 0.63 & -17.21 & -13.43 & 0 & \dots & 0 & 0 & & & & \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \\ 2.73 & -0.56 & 0.87 & 0 & 0 & 0 & \dots & -3.73 & 7.93 \cdot 10^4 & 2.44 & 0 & \dots & -7.93 \cdot 10^4 \\ 5.66 & -0.60 & 4.16 & 0 & 0 & 0 & \dots & -5.66 & 0.31 & -6.43 & 13.75 & \dots & 0 \\ 0.82 & -0.29 & 1.98 & 0 & 0 & 0 & \dots & -0.82 & 0.11 & -3.06 & 6.55 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & -0.01 & 0.11 & 0.63 & -17.22 & 13.43 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} \Delta M_k \\ \Delta P_k \\ \Delta h_k \\ \Delta \Theta_k^j \\ \dots \end{bmatrix} \quad (58)$$

Model rur ekranowych parownika opisuje wolne procesy cieplne (przewodnictwo cieplne wzdłuż elementów metalowych – ścianki rur, płetwy łączące), procesy transportu masy i zmian entalpii czynnika roboczego (wody, mieszaniny parowo-wodnej) wzdłuż rur ekranowych, a także bardzo szybkie procesy dynamiczne związane z przemieszczaniem się zmian ciśnienia i przepływu masowego wzdłuż rur ekranowych (procesy dźwiękowe). Z tego powodu wartości własne zlinearyzowanego modelu rur ekranowych o parametrach skupionych posiadają wartości różniące się o kilka rzędów (rys. 19):

- wartości własne związane z przewodnictwem cieplnym (linia niebieska),
- z transportem masy (linia zielona),
- z przemieszczaniem się zmian ciśnienia (linia czerwona).

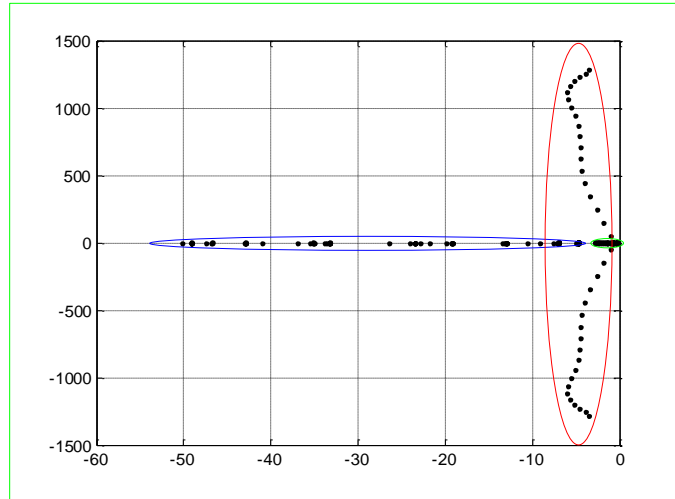
Ponieważ ze zjawiskami dźwiękowymi są związane silne i szybkie oscylacje ciśnienia i przepływu masowego, wartości własne odpowiadające tym procesom przyjmują wartości zespolone).

Przy wartościach własnych różniących się o dwa, a nawet o trzy rzędy, procesy dynamiczne związane z przemieszczaniem się zmian ciśnienia wzdłuż rur ekranowych odbywają się praktycznie niezależnie od pozostałych procesów termodynamicznych. Z tego powodu w modelu rur ekranowych parownika można wyróżnić dwa podsystemy słabo ze sobą powiązane:

- model opisujący szybkie zjawiska dźwiękowe,
- model opisujący wolne zjawiska cieplne i termodynamiczne.

Po wykonaniu standaryzacji zgodnie z algorytmem dekompozycji Epsilon, fragment binarnej macierzy połączeń dla równań stanu opisujących dwa sąsiednie elementy skończone zawierające po 15 zmiennych stanu, dla $\varepsilon=0.05$ jest przedstawiony na rys. 20. Silne powiązania między zmiennymi M_k i P_k są traktowane tak samo, jak znacznie słabsze powiązania między zmiennymi M_k , P_k , h_k oraz Θ_k . Występujące powiązania nie pozwalają algorytmowi dekompozycji Epsilon na wyodrębnienie podsystemów (funkcja *Connected_Components* zwraca 1 blok).

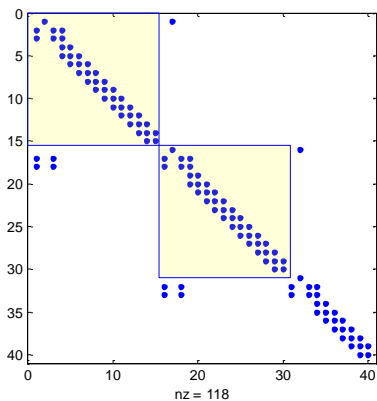
W pracy zaproponowano zmodyfikowany algorytm dekompozycji Epsilon, pozwalający na wyodrębnienie silnie związanych zmiennych stanu, którym odpowiadają szybkie oscylacyjne procesy dynamiczne. Algorytm pozwala na dekompozycję slow-fast modelu złożonego, w którym szybkie procesy dynamiczne mają charakter oscylacyjny o wysokiej częstotliwości. Z taką sytuacją mamy do czynienia w przypadku modeli podsystemów rur ekranowych parownika.



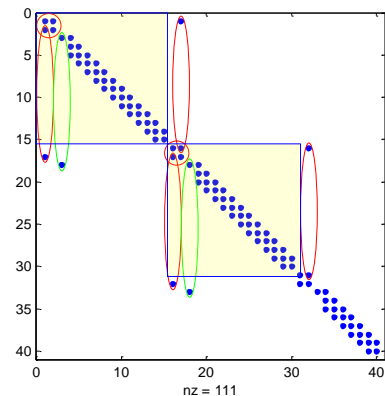
RYSUNEK 19. WARTOŚCI WŁASNE MODELU STREFY DOGRZEWU DLA 20 ELEMENTÓW SKOŃCZONYCH ORAZ RZĘDU ELEMENTU SKOŃCZONEGO 15

Modyfikacja algorytmu dekompozycji Epsilon polega na takim skonstruowaniu binarnej macierzy połączeń, aby procedura *Connected_Components* biblioteki BOOST, mogła wyodrębnić podsystem silnie powiązanych równań stanu opisujących szybkie procesy oscylacyjne oraz podsystem zawierający równania stanu opisujące wolne procesy cieplne.

W celu zidentyfikowania równań stanu silnie powiązanych ze sobą, analizowane są wszystkie pary równań stanu (przy założeniu, że dana para jest odseparowana od pozostałych równań stanu), w celu wyznaczenia dwóch wartości własnych i sprawdzenia, czy posiadają dużą wartość urojoną. Jeżeli tak, to w binarnej macierzy połączeń zostają wpisane w odpowiednich wierszach i kolumnach wartości 1 (analizowane zmienne stanu są ze sobą silnie powiązane).



RYSUNEK 20. BINARNA MACIERZ POŁĄCZEŃ DLA MODELU STREFY DOGRZEWU, UZYSKANA Z ZASTOSOWANIEM KLASYCZNEGO ALGORYTMU DEKOMPOZYCJI EPSILON



RYSUNEK 21. BINARNA MACIERZ POŁĄCZEŃ DLA MODELU STREFY DOGRZEWU PAROWNIKA, UZYSKANA NA PODSTAWIE ZMODYFIKOWANEGO ALGORYTMU DEKOMPOZYCJI EPSILON

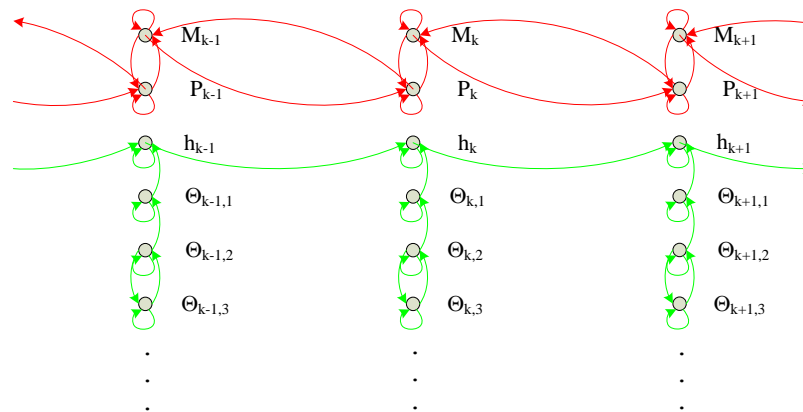
Wartości własne odpowiadające zmiennym stanu x_k oraz x_l (przy założeniu, że dwa równania stanu są odseparowane od pozostałych równań) można wyznaczyć następująco:

$$\frac{d}{dt} \begin{bmatrix} \dots \\ x_k \\ \dots \\ x_l \\ \dots \end{bmatrix} = \begin{bmatrix} \dots & \dots & \dots \\ \dots & a_{kk} & \dots & a_{kl} & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & a_{lk} & \dots & a_{ll} & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} \dots \\ x_k \\ \dots \\ x_l \\ \dots \end{bmatrix}, s_{1,2} \cong \frac{a_{kk} + a_{ll}}{2} \pm j \sqrt{|a_{kl}a_{lk}|} \text{ dla } |a_{kl}a_{lk}| \gg 1 \text{ oraz } a_{kl}a_{lk} < 0. \quad (59)$$

Pozostałe powiązania zmiennych stanu zostają wyznaczone zgodnie z algorytmem dekompozycji Epsilon. Binarną mapę połączeń uzyskaną na podstawie zmodyfikowanego algorytmu Epsilon przedstawiono na rys. 21. Zastosowanie procedury *Connected_Components* dla uzyskanej macierzy połączeń daje dwa podsystemy, z których pierwszy charakteryzuje się dużymi wartościami własnymi i opisuje szybkie dynamiczne procesy oscylacyjne, natomiast drugi opisuje wolne procesy cieplne.

Przedstawiony algorytm zmodyfikowanej dekompozycji Epsilon sprawdza się dla modeli, w których występują silne oscylacje, a jednocześnie wolnozmiennie procesy dynamiczne. Podobnie jak dekompozycja Epsilon, przedstawiony algorytm jest algorytmem heurystycznym, nie zapewniającym, że dokonana dekompozycja spełnia jednoznaczne kryteria. W algorytmie należy eksperymentalnie dobrać wartość ε oraz wartość Q_{min} , która oznacza minimalną wartość zespoloną dla wartości własnych, dla których przyjmuje się, że zmienne stanu są silnie powiązane.

Na rys 22 przedstawiono powiązania między dwoma podsystemami (linia czerwona – podsystem szybki, linia zielona – podsystem wolny). Z rys. 22 wynika, że podsystemy są odseparowane od siebie.



RYSUNEK 22. GRAF POWIĄZAŃ ZMIENNYCH STANU DLA ZMODYFIKOWANEJ DEKOMPOZYCJI EPSILON. (kolorem czerwonym zaznaczono podsystem z dużymi wartościami bezwzględnych wartości własnych (szybkie procesy dynamiczne), natomiast kolorem zielonym – podsystem wolnych procesów dynamicznych)

W wyniku dekompozycji modelu na część szybką (zespolone wartości własne o dużej wartości bezwzględnej) oraz część wolną (rzeczywiste wartości własne o małej wartości bezwzględnej) uzyskuje się dwa podsystemy odseparowane od siebie w znacznym stopniu. Realizację algorytmu wyznaczania binarnej macierzy połączeń dla zmodyfikowanego algorytmu Epsilon przedstawiono w postaci programu w środowisku MATLAB (rys. 23).

3.6. Dekompozycja Multi-Level Substructuring (MLS)

Dekompozycja MLS (Multi-Level Substructuring) została opisana w pracach Franka Blömelinga [31] [123]. Umożliwia ona wyznaczenie dekompozycji modelu na dwie części, które dla stanu ustalonego sumarycznie generują taką samą odpowiedź jak model pierwotny. Dla tego rodzaju dekompozycji wprowadzono następujące macierze transformacji:

$$R = \begin{bmatrix} I & -A_{11}^{-1}A_{12} \\ \emptyset & I \end{bmatrix}, \quad F = \begin{bmatrix} I & \emptyset \\ -A_{21}A_{11}^{-1} & I \end{bmatrix}, \quad (60)$$

w celu realizacji dekompozycji modelu na dwie części.

$$\hat{A} = FAR, \quad \hat{B} = FB, \quad \hat{C} = CR, \quad \hat{E} = FR, \quad (61)$$

$$\hat{A} = \begin{bmatrix} A_{11} & \emptyset \\ \emptyset & \bar{A}_{22} \end{bmatrix}, \quad \hat{E} = \begin{bmatrix} I & \bar{E}_{12} \\ \bar{E}_{21} & \bar{E}_{22} \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} B_1 \\ \bar{B}_2 \end{bmatrix}, \quad \hat{C} = [C_1 \quad \bar{C}_2], \quad (62)$$

$$\bar{A}_{22} = A_{22} - A_{21}A_{11}^{-1}A_{12}, \quad (63)$$

$$\bar{E}_{12} = -A_{11}^{-1}A_{12}, \quad \bar{E}_{21} = -A_{21}A_{11}^{-1}, \quad \bar{E}_{22} = I + A_{21}A_{11}^{-1}A_{11}^{-1}A_{12}, \quad (64)$$

$$\bar{B}_2 = B_2 - A_{21}A_{11}^{-1}B_1, \quad \bar{C}_2 = C_2 - C_1A_{11}^{-1}A_{12}. \quad (65)$$

W wyniku przekształceń otrzymany zostaje model z blokowo-diagonalną strukturą macierzy \hat{A} , jednak transformacja przestrzeni stanu powoduje przekształcenie macierzy E (będącej macierzą jednostkową dla modelu oryginalnego) do postaci bardziej złożonej.

Redukcja otrzymanych w wyniku dekompozycji modeli odbywa się niezależnie z pominięciem bloków macierzy \hat{E} leżących poza główną przekątną (interfejsy pomiędzy modelami) [31]. Równania stanu modelu:

$$\begin{bmatrix} I & \bar{E}_{12} \\ \bar{E}_{21} & \bar{E}_{22} \end{bmatrix} \begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & \emptyset \\ \emptyset & \bar{A}_{22} \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ \bar{B}_2 \end{bmatrix} u \quad y = [C_1 \quad \bar{C}_2] \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}, \quad (66)$$

zostają doprowadzone zatem do postaci:

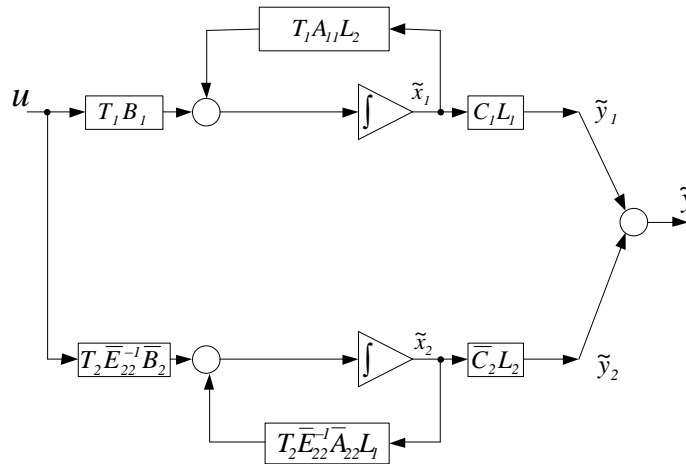
$$\begin{bmatrix} I & \emptyset \\ \emptyset & \bar{E}_{22} \end{bmatrix} \begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & \emptyset \\ \emptyset & \bar{A}_{22} \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ \bar{B}_2 \end{bmatrix} u \quad y = [C_1 \quad \bar{C}_2] \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix}, \quad (67)$$

Po lewostronnym pomnożeniu równań związanych z wektorem stanu \hat{x}_2 przez \bar{E}_{22}^{-1} otrzymane zostają równania:

$$\begin{cases} \dot{\hat{x}}_1 = A_{11}\hat{x}_1 + B_1u \\ y_1 = C_1\hat{x}_1 \end{cases}, \quad \begin{cases} \dot{\hat{x}}_2 = \bar{E}_{22}^{-1}\bar{A}_{22}\hat{x}_2 + \bar{E}_{22}^{-1}\bar{B}_2u \\ y_2 = \bar{C}_2\hat{x}_2 \end{cases} \quad (68)$$

W wyniku procesu redukcji wyznaczone zostają macierze transformacji T_1 , L_1 oraz T_2 , L_2 . Łączenie zredukowanych podsystemów zostaje zrealizowane zgodnie ze schematem przedstawionym na rysunku 25.

Dekompozycja MLS wykazuje dobre właściwości w przypadku niesymetrycznego podziału macierzy stanu (zmieniona w wyniku przekształceń część modelu powinna posiadać znacznie mniejszy rozmiar od części nie podlegającej modyfikacji). Wydzielone w wyniku dekompozycji MLS podsystemy nie są ze sobą powiązane żadnym interfejsem.



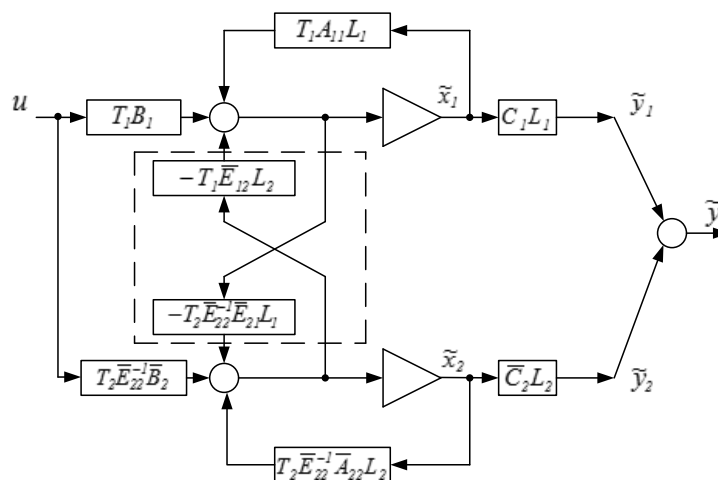
RYSUNEK 25. SCHEMAT ŁĄCZENIA MODELI ZREDUKOWANYCH W WYNIKU DEKOMPOZYCJI MLS

3.7. Dekompozycja oparta na dopełnieniu Schur'a - SchComp (ang. Schur complement)

Algorytm redukcji z wykorzystaniem dekompozycji MLS prezentowany w pracach Blömelinga pomija połączenia wydzielonych podsystemów obecne w macierzy \hat{E} . W pracy zaprezentowano modyfikację algorytmu dekompozycji opracowanego przez Blömelinga, która wykorzystuje dopełnienie Schura w celu wydzielenia dwóch podsystemów z uwzględnieniem bloków \bar{E}_{12} oraz \bar{E}_{21} obecnych w macierzy \hat{E} . Uwzględnienie obu interfejsów między zdekomponowanymi modelami, prowadzi do otrzymania równań stanu w postaci:

$$\begin{cases} \dot{\hat{x}}_1 = A_{11}\hat{x}_1 + B_1u - \bar{E}_{12}\dot{\hat{x}}_2 \\ y_1 = C_1\hat{x}_1 \end{cases}, \quad \begin{cases} \dot{\hat{x}}_2 = \bar{E}_{22}^{-1}\bar{A}_{22}\hat{x}_2 + \bar{E}_{22}^{-1}\bar{B}_2u - \bar{E}_{22}^{-1}\bar{E}_{21}\dot{\hat{x}}_1 \\ y_2 = \bar{C}_2\hat{x}_2 \end{cases} \quad (69)$$

Podobnie jak w przypadku algorytmu prezentowanego w pracach Blömelinga, redukcji podlegają modele określone zależnościami (68). Natomiast wyznaczone macierze transformacji zostają użyte do przekształcenia macierzy interfejsów \bar{E}_{12} oraz \bar{E}_{21} . Schemat łączenia zredukowanych części został przedstawiony na rysunku 26.



RYSUNEK 26. SCHEMAT ŁĄCZENIA MODELI ZREDUKOWANYCH W WYNIKU DEKOMPOZYCJI SCHUR COMPLEMENT

Zredukowany i połączony model może zostać zatem przedstawiony w postaci równań macierzowych jako:

$$\begin{bmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \end{bmatrix} = \begin{bmatrix} I & \tilde{E}_{12} \\ \tilde{E}_{21} & I \end{bmatrix}^{-1} \begin{bmatrix} \tilde{A}_{11} & \emptyset \\ \emptyset & \tilde{A}_{22} \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} + \begin{bmatrix} I & \tilde{E}_{12} \\ \tilde{E}_{21} & I \end{bmatrix}^{-1} \begin{bmatrix} \tilde{B}_1 \\ \tilde{B}_2 \end{bmatrix} u, \quad \tilde{y} = \begin{bmatrix} \tilde{C}_1 & \tilde{C}_2 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} \quad (70)$$

gdzie:

$$\tilde{A}_{11} = T_1 A_{11} L_1, \quad \tilde{A}_{22} = T_2 \bar{E}_{22}^{-1} \bar{A}_{22} L_2 \quad (71)$$

$$\tilde{B}_1 = T_1 B_1, \quad \tilde{B}_2 = T_2 \bar{E}_{22}^{-1} \bar{B}_2 \quad (72)$$

$$\tilde{E}_{12} = T_1 \bar{E}_{12} L_2, \quad \tilde{E}_{21} = T_2 \bar{E}_{22}^{-1} \bar{E}_{21} L_1 \quad (73)$$

$$\tilde{C}_1 = C_1 L_1, \quad \tilde{C}_2 = \bar{C}_2 L_2. \quad (74)$$

3.8. Zmodyfikowana dekompozycja oparta na dopełnieniu Schur'a (SchCompZm)

W poprzedniej metodzie realizacja interfejsów między wydzielonymi podsystemami realizowana była poprzez macierze \bar{E}_{12} i \bar{E}_{21} oraz pochodne zmiennych stanu. Równania (69) mogą zostać przeformułowane w taki sposób, aby interfejsy pomiędzy modelami wydzielonych podsystemów oparte były na zmiennych stanu.

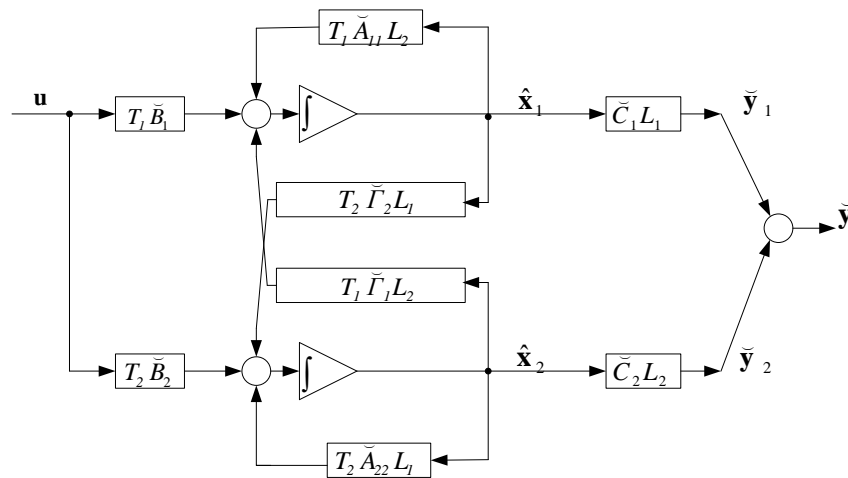
Po przekształceniach uzyskuje się następujące modele podsystemów wraz z interfejsami:

$$\dot{\hat{x}}_1 = \bar{A}_{11} \hat{x}_1 + \bar{B}_1 u - \bar{\Gamma}_1 \hat{x}_2, \quad \dot{\hat{x}}_2 = \bar{A}_{22} \hat{x}_2 + \bar{B}_2 u - \bar{\Gamma}_2 \hat{x}_1 \quad (75)$$

gdzie:

$$\begin{aligned} \bar{A}_{11} &= (I - \bar{E}_{12} \bar{E}_{22}^{-1} \bar{E}_{21})^{-1} A_{11} & \bar{A}_{22} &= (I - \bar{E}_{22}^{-1} \bar{E}_{21} \bar{E}_{12})^{-1} \bar{E}_{22}^{-1} \bar{A}_{22} \\ \bar{B}_1 &= (I - \bar{E}_{12} \bar{E}_{22}^{-1} \bar{E}_{21})^{-1} (B_1 - \bar{E}_{12} \bar{E}_{22}^{-1} \bar{B}_2) & \bar{B}_2 &= (I - \bar{E}_{22}^{-1} \bar{E}_{21} \bar{E}_{12})^{-1} \bar{E}_{22}^{-1} (\bar{B}_2 - \bar{E}_{21} B_1) \\ \bar{\Gamma}_1 &= (I - \bar{E}_{12} \bar{E}_{22}^{-1} \bar{E}_{21})^{-1} \bar{E}_{12} \bar{E}_{22}^{-1} \bar{A}_{22} & \bar{\Gamma}_2 &= (I - \bar{E}_{22}^{-1} \bar{E}_{21} \bar{E}_{12})^{-1} \bar{E}_{22}^{-1} \bar{E}_{21} A_{11} \end{aligned} \quad (76)$$

Analogicznie jak poprzednio, wydzielone modele podsystemów są redukowane niezależnie, bez uwzględniania interfejsów. Po redukcji interfejsy $\bar{\Gamma}_1$ i $\bar{\Gamma}_2$ są transformowane z wykorzystaniem wyznaczonych macierzy transformacji T_1, L_1, T_2, L_2 (rys. 27).



RYСУNEK 27. SCHEMAT ŁĄCZENIA MODELI ZREDUKOWANYCH Z ZASTOSOWANIEM ZMODYFIKOWANEJ METODY SCHUR COMPLEMENT (SchCompZm)

3.9. Dekompozycja Schur'a (Sch)

Rzeczywista dekompozycja Schur'a macierzy A może zostać zapisana następująco [124]:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = U\tilde{A}U^T = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ u_{n1} & u_{n2} & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} \tilde{a}_{11} & \tilde{a}_{12} & \cdots & \tilde{a}_{1n} \\ \tilde{a}_{21} & \ddots & & \vdots \\ 0 & \ddots & \ddots & \vdots \\ 0 & 0 & \tilde{a}_{n(n-1)} & \tilde{a}_{nn} \end{bmatrix} \begin{bmatrix} u_{11} & u_{21} & \cdots & u_{n1} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ u_{1n} & u_{2n} & \cdots & u_{nn} \end{bmatrix} \quad (77)$$

Gdzie macierz \tilde{A} posiada strukturę quasi-trójkątną górną z blokami o rozmiarze 2×2 znajdującymi się na przekątnej. Niezerowe elementy poniżej przekątnej związane są z zespolonymi wartościami własnymi macierzy A . Macierz U jest macierzą ortogonalną. Podział modelu oryginalnego na modele dwóch podsystemów może zostać wykonany w następujący sposób. W pierwszej kolejności wykonywana jest transformacja macierzy stanu modelu zgodnie z zależnością:

$$\tilde{A} = U^T A U, \quad \tilde{B} = U^T B, \quad \tilde{C} = C U \quad (78)$$

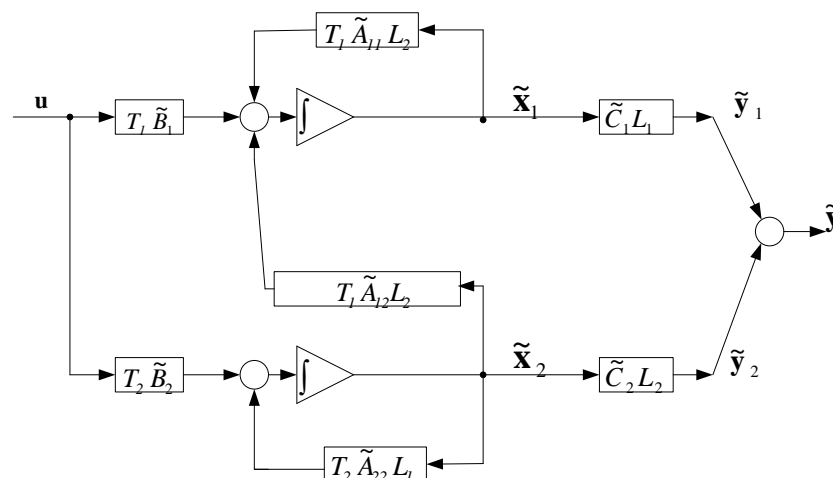
Po wykonaniu przekształcenia, struktura macierzy A modelu zostaje doprowadzona do postaci quasi-trójkątnej górnej. Podział modelu na dwie części realizowany jest w taki sposób, aby blok \tilde{A}_{21} był macierzą zerową (podział nie może rozdzielić bloku należącego do zespolonych wartości własnych macierzy A):

$$\tilde{A} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \emptyset & \tilde{A}_{22} \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} \tilde{B}_1 \\ \tilde{B}_2 \end{bmatrix}, \quad \tilde{C} = [\tilde{C}_1 \quad \tilde{C}_2] \quad (79)$$

Oznacza to, że wydzielone bloki w macierzy \tilde{A} mogą mieć różne rozmiary. Otrzymane dwa modele podsystemów mogą zostać zapisane za pomocą równań:

$$\begin{aligned} \dot{\tilde{x}}_1 &= \tilde{A}_{11}\tilde{x}_1 + \tilde{B}_1 u + \tilde{A}_{12}\tilde{x}_2, & \tilde{y}_1 &= \tilde{C}_1\tilde{x}_1 \\ \dot{\tilde{x}}_2 &= \tilde{A}_{22}\tilde{x}_2 + \tilde{B}_2 u, & \tilde{y}_2 &= \tilde{C}_2\tilde{x}_2 \end{aligned} \quad (80)$$

Obydwa modele podsystemów są następnie redukowane niezależnie. Wyznaczone w ten sposób macierze transformacji T_1, L_1, T_2, L_2 wykorzystywane są do wyznaczania macierzy interfejsu: $T_1\tilde{A}_{12}L_2$ (rys. 28).



RYSUNEK 28. SCHEMAT ŁĄCZENIA MODELI ZREDUKOWANYCH W WYNIKU DEKOMPOZYCJI SCHUR'A

3.10. Dekompozycja Schur-Sylvester (SchSy1v)

Dekompozycja Schur-Sylvester umożliwia wydzielenie z modelu oryginalnego dwóch modeli podsystemów, które są od siebie całkowicie odseparowane (posiadają zerowe macierze interfejsów). Realizowane jest to poprzez wykonanie dwóch transformacji macierzy stanu modelu oryginalnego. Pierwsze przekształcenie, podobnie jak w przypadku dekompozycji Schura, polega na doprowadzeniu macierzy A modelu oryginalnego do postaci (77). Następnie wyznaczana jest macierz transformacji Z i jej odwrotność Z^{-1} :

$$Z = \begin{bmatrix} I & X \\ \emptyset & I \end{bmatrix}, \quad Z^{-1} = \begin{bmatrix} I & -X \\ \emptyset & I \end{bmatrix} \quad (81)$$

tak, aby doprowadzić macierz $\tilde{\mathbf{A}}$ do postaci blokowo-diagonalnej:

$$Z^{-1}\tilde{\mathbf{A}}Z = \begin{bmatrix} I & -X \\ \emptyset & I \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{A}}_{11} & \tilde{\mathbf{A}}_{12} \\ \emptyset & \tilde{\mathbf{A}}_{22} \end{bmatrix} \begin{bmatrix} I & X \\ \emptyset & I \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{A}}_{11} & \emptyset \\ \emptyset & \hat{\mathbf{A}}_{22} \end{bmatrix} \quad (82)$$

Macierz X wyznaczana jest poprzez rozwiązanie równania Sylvester'a:

$$\tilde{\mathbf{A}}_{11}X - X\tilde{\mathbf{A}}_{22} + \tilde{\mathbf{A}}_{12} = 0 \quad (83)$$

które wynika z wyzerowania macierzy $\tilde{\mathbf{A}}_{12}$:

$$Z^{-1}\tilde{\mathbf{A}}Z = \begin{bmatrix} I & -X \\ \emptyset & I \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{A}}_{11} & \tilde{\mathbf{A}}_{12} \\ \emptyset & \tilde{\mathbf{A}}_{22} \end{bmatrix} \begin{bmatrix} I & X \\ \emptyset & I \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{A}}_{11} & \tilde{\mathbf{A}}_{11}X + \tilde{\mathbf{A}}_{12} - X\tilde{\mathbf{A}}_{22} \\ \emptyset & \tilde{\mathbf{A}}_{22} \end{bmatrix} \quad (84)$$

Uzyskane macierze transformacji służą do uzyskania dwóch podsystemów:

$$\hat{\mathbf{A}} = \begin{bmatrix} I & -X \\ \emptyset & I \end{bmatrix} \tilde{\mathbf{A}} \begin{bmatrix} I & X \\ \emptyset & I \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{A}}_{11} & \emptyset \\ \emptyset & \hat{\mathbf{A}}_{22} \end{bmatrix} \Rightarrow \begin{matrix} \hat{\mathbf{A}}_1 = \hat{\mathbf{A}}_{11} \\ \hat{\mathbf{A}}_2 = \hat{\mathbf{A}}_{22} \end{matrix} \quad (85)$$

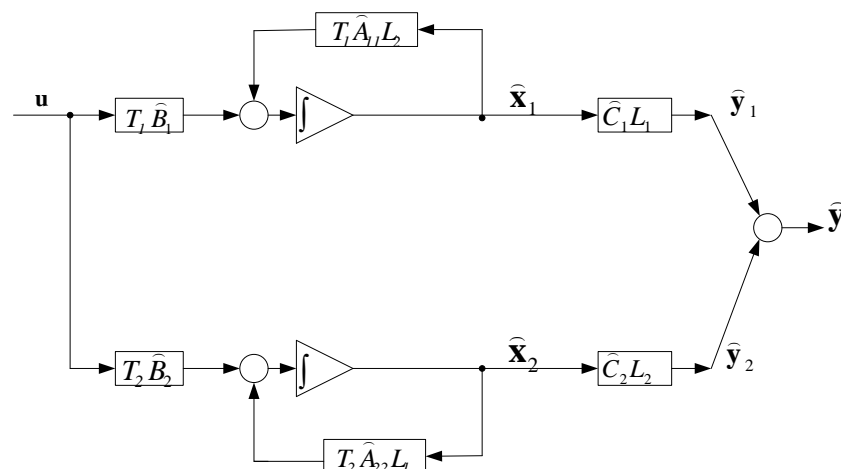
$$C \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} I & X \\ \emptyset & I \end{bmatrix} = C \begin{bmatrix} U_1 & U_1X + U_2 \end{bmatrix} \Rightarrow \begin{matrix} \hat{C}_1 = CU_1 \\ \hat{C}_2 = CU_1X + CU_2 \end{matrix} \quad (86)$$

$$\begin{bmatrix} I & -X \\ \emptyset & I \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} B = \begin{bmatrix} V_1 - XV_2 \\ V_2 \end{bmatrix} B \Rightarrow \begin{matrix} \hat{B}_1 = V_1B - XV_2B \\ \hat{B}_2 = V_2B \end{matrix} \quad (87)$$

gdzie :

$$V = U^T \quad (88)$$

Uzyskane modele pozostają niezależne od siebie, mogą zostać zredukowane i połączone zgodnie ze schematem 29.



RYSUNEK 29. SHEMAT ŁĄCZENIA MODELI ZREDUKOWANYCH W WYNIKU DEKOMPOZYCJI SCHUR-SYLVESTER

3.11. Uwagi

Dekompozycja modelu systemu na modele podsystemów daje następujące możliwości:

- redukcja modeli wysokiego rzędu, których nie można zredukować w jednym kroku ze względu na długi czas obliczeń oraz rozmiar pamięci operacyjnej,
- zmniejszenie złożoności obliczeniowej redukcji modelu wysokiego rzędu, gdyż przy dekompozycji modelu na m podsystemów złożoność obliczeniowa wynosi

$$O\left(m\left(\frac{n}{m}\right)^3\right)$$

- możliwość zastosowania obliczeń równoległych, gdyż każdy zdekomponowany podsystem może być redukowany niezależnie od pozostałych.

Operacja scalania modelu na podstawie zredukowanych modeli podsystemów jest operacją znacznie mniej czasochłonną od operacji redukcji.

Redukcja z użyciem dekompozycji strukturalnej jest najbardziej naturalna, gdyż wydziela określone liczby elementów skończonych do każdego podsystemu. Ten rodzaj dekompozycji może być stosowany do modeli, w których interfejsy $\Gamma_{1,2}$ i $\Gamma_{2,1}$ zawierają niewielką liczbę elementów niezerowych (szczególnie dla metody FW, w której dla każdego wejścia/wyjścia dołączony jest filtr o parametrach wyznaczanych z zastosowaniem algorytmu ewolucyjnego). Dodatkowo dekompozycja strukturalna nie zmienia gęstości macierzy stanu – w przypadku gdy dekompozycji poddano system opisany macierzami rzadkimi, zdekomponowane podsystemy również będą opisane macierzami stanu o postaci rzadkiej.

Redukcja z użyciem dekompozycji strukturalnej uproszczonej pomija interfejsy $\Gamma_{1,2}$ i $\Gamma_{2,1}$ w procesie redukcji, lecz w wyniku daje modele o większym błędzie aproksymacji charakterystyk częstotliwościowych. Ponadto w wyniku dekompozycji strukturalnej uproszczonej można uzyskać podsystemy niesterowalne/nieobserwowalne, co powoduje, że redukcja modeli podsystemów staje się niemożliwa. Podobnie jak dekompozycja strukturalna, w wyniku dekompozycji strukturalnej uproszczonej gęstość macierzy stanu nie ulega zmianie.

Szczególnie interesująca jest dekompozycja slow-fast. Uzyskane podsystemy są w znacznym stopniu niezależne (systemy singularne, w których procesy dynamiczne przebiegają w dwóch różnych skalach czasowych). W pracy zastosowano dekompozycję Epsilon do wydzielenia podsystemów szybkiego i wolnego, dla przypadku gdy w modelu występują szybkie procesy oscylacyjne. Po wyznaczeniu binarnej macierzy połączeń oraz zastosowaniu procedury `connected_components` biblioteki BOOST, w wyniku permutacji macierzy uzyskuje się dwa podsystemy o różnych skalach czasowych.

Operację wydzielenia podsystemów w dekompozycji slow-fast można zrealizować z zastosowaniem poszczególnych metod dekompozycji, w ten sposób powstaje cały szereg metod dekompozycji slow-fast: *Esf-Str*, *Esf-StrUpr*, *Esf-SchSylv*, *Esf-SchComp*, *Esf-Sch*, *Esf-MLS*, *Esf-SchCompZm*.

Dekompozycja MLS charakteryzuje się zerowym błędem modelu zdekomponowanego dla stanu ustalonego. W metodzie pomija się interfejs między wydzielonymi podsystemami, na skutek czego już na poziomie dekompozycji zostaje wprowadzony błąd modelu, który ujawnia się dla stanów nieustalonych. Dekompozycja MLS powinna zostać wykonana w taki sposób, aby rozmiar modelu podsystemu numer 1 z równania (68) (część nie ulegająca zmianie w wyniku dekompozycji) był znacznie większy od rozmiaru modelu podsystemu numer 2 z równania (68) (część modyfikowana lewostronnym mnożeniem przez \bar{E}_{22}^{-1}). Dla modeli analizowanych w pracy, metoda MLS może być zastosowana w dekompozycji slow-fast (*Esf-MLS*).

W metodach Schur Complement (*SchComp*) oraz zmodyfikowanej metodzie Schur Complement (*SchCompZm*), zaproponowanych przez autora, uwzględnia się interfejs między podsystemami uzyskanymi w metodzie MLS. W związku z tym sama dekompozycja nie wprowadza błędów dla stanów nieustalonych. Redukcji podlegają, podobnie jak to było dla dekompozycji MLS, modele określone równaniami (68). Wyznaczone macierze transformacji są wykorzystywane do przekształcenia interfejsów znajdujących się w macierzy E.

Znana z algebry dekompozycja Schur'a (*Sch*) daje możliwość wydzielenia dwóch podsystemów bez sprzężenia zwrotnego (jeden z interfejsów jest zerowy). Kosztem tego zostaje wprowadzony jeden gęsty interfejs umieszczony w bloku A_{J_2} macierzy stanu. Natomiast dla dekompozycji Schur-Sylvester (*SchSylv*) uzyskuje się dwa niezależne podsystemy połączone równolegle. W celu realizacji tej dekompozycji należy rozwiązać równanie macierzowe Sylvester'a, które charakteryzuje się złożonością obliczeniową $O(n^3)$.

Zaproponowane metody dekompozycji modeli LTI można porównać pod różnymi względami:

- stabilność modelu zredukowanego uzyskanego na podstawie połączenia zredukowanych modeli podsystemów,
- błąd aproksymacji charakterystyk częstotliwościowych przez model zredukowany w zadanym przedziale częstotliwości,
- złożoność obliczeniowa (czas obliczeń związany z redukcją modeli podsystemów oraz scalaniem modeli zredukowanych).

Dla stabilnych modeli zredukowanych, w przypadku użycia dekompozycji nie eliminujących choćby jednego z interfejsów, w wyniku scalania możliwe jest uzyskanie modelu niestabilnego. Taka sytuacja może wystąpić w następujących przypadkach dekompozycji: *Str*, *StrUpr*, *Esf-Str*, *Esf-StrUpr*, *Esf-SchComp*, *Esf-SchCompZm*, *SchComp*, *SchCompZm*. Metody *Sch*, *SchSylv*, *MLS* oraz *Esf-SchSylv*, *Esf-Sch*, *Efs-MLS* charakteryzują się brakiem utraty stabilności modelu w wyniku procesu scalania (brak sprzężeń zwrotnych między podsystemami).

Każda z metod dekompozycji charakteryzuje się innym błędem aproksymacji. W pracy porównuje się błędy średniokwadratowe względne charakterystyk częstotliwościowych w zadanym przedziale częstotliwości dla określonej liczby punktów.

Podczas redukcji modeli wysokiego rzędu istotnym parametrem jest czas obliczeń, związany ze złożonością obliczeniową. Przy obliczeniach równoległych zdekomponowane modele powinny mieć zbliżony rozmiar, aby operacje redukcji wykonać równolegle z zastosowaniem wielu węzłów klastra komputerowego. Dla modeli analizowanych w pracy, zastosowanie dekompozycji slow-fast daje modele podsystemów o znacznie różniących się rozmiarach. Oznacza to, że dekompozycja slow-fast mimo szeregu zalet (stabilność modeli zredukowanych, małe błędy aproksymacji) pociąga za sobą długie czasy obliczeń.

4. Metody wielopoziomowej dekompozycji modeli LTI

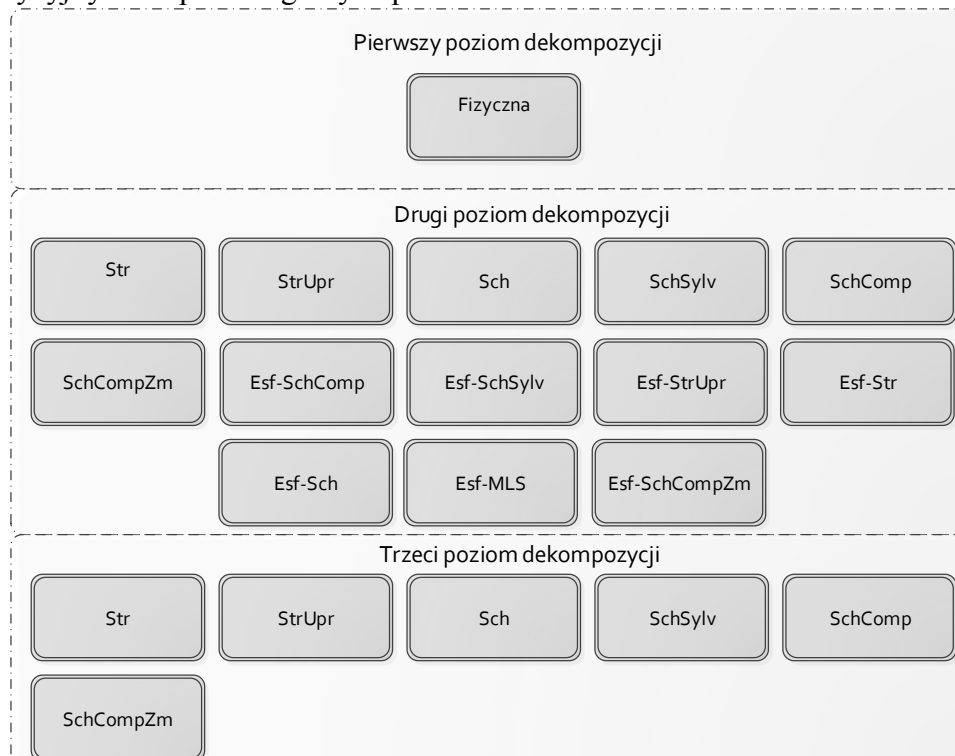
4.1. Dekompozycja wielopoziomowa

W rozdziale 3 zostały opisane podstawowe metody dekompozycji modeli zapisanych w postaci równań w przestrzeni stanu. Metody dekompozycji: fizyczna, strukturalna i strukturalna uproszczona, Schur'a, Schur-Sylvester, Schur complement, Schur complement zmodyfikowana oraz Epsilon slow-fast stanowią metody umożliwiające rozdzielenie modelu na dwa podsystemy. Optymalnym rozwiązaniem w przypadku tych metod dekompozycji jest podział modelu na podsystemy o zbliżonych rozmiarach (ze względu na zbliżoną złożoność obliczeniową procesu redukcji podczas obliczeń równoległych).

Ze względu na wysoką złożoność analizowanego modelu parownika kotła energetycznego BP-1150, w pracy zastosowano wielopoziomową dekompozycję modelu:

- Pierwszy poziom dekompozycji stanowi dekompozycja fizyczna, w wyniku której uzyskiwane są modele poszczególnych podsystemów parownika. Ich wzajemne połączenia realizowane poprzez wielkości wejściowe i wyjściowe podsystemów (w tym także sprzężenia zwrotne), wspólnie definiują model całości.
- Kolejne poziomy dekompozycji wprowadzone zostają w celu zmniejszenia złożoności obliczeniowej procesu redukcji poszczególnych podsystemów. Z praktycznego punktu widzenia, ze względu na złożoność obliczeniową metod redukcji SVD wykorzystywanych w metodzie FW oraz znaczną liczbę przebiegów algorytmu ewolucyjnego przyjęto, że model poddawany redukcji powinien mieć około 10^3 zmiennych stanu.

Na rysunku 30 przedstawiono rozważane kombinacje wykorzystania metod dekompozycyjnych na poszczególnych poziomach.



RYСУNEK 30. ROZWAŻANE KOMBINACJE DLA WIELOPOZIOMOWYCH METOD DEKOMPOZYCYJNYCH

4.2. Realizacja dekompozycji wielopoziomowej z redukcją modelu

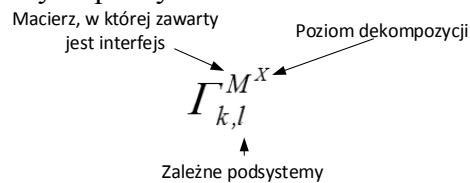
Dla modeli zapisanych w postaci macierzowych równań stanu ogólny schemat dwupoziomowej dekompozycji modelu został zilustrowany na rysunku 31.

$$\begin{array}{|c|c|c|c|} \hline E_1^2 & \Gamma_{2,1}^{E^2} & & \dot{x}_1^2 \\ \hline \Gamma_{1,2}^{E^2} & E_2^2 & & \dot{x}_2^2 \\ \hline & & \Gamma_{2,1}^{E^1} & \dot{x}_1^1 \\ \hline & & & \dot{x}_2^1 \\ \hline & & & \dot{x}_3^2 \\ \hline & & & \dot{x}_4^2 \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline A_1^2 & \Gamma_{2,1}^{A^2} & & x_1^2 \\ \hline \Gamma_{1,2}^{A^2} & A_2^2 & & x_2^2 \\ \hline & & \Gamma_{2,1}^{A^1} & x_1^1 \\ \hline & & & x_2^1 \\ \hline & & & x_3^2 \\ \hline & & & x_4^2 \\ \hline \end{array} + \begin{array}{|c|} \hline B_1^2 \\ \hline B_2^2 \\ \hline B_3^2 \\ \hline B_4^2 \\ \hline \end{array} \vec{u}$$

$$\mathbf{y} = \begin{array}{|c|c|c|c|} \hline C_1^2 & C_2^2 & C_3^2 & C_4^2 \\ \hline \end{array} \begin{array}{|c|} \hline x_1^2 \\ \hline x_2^2 \\ \hline x_3^2 \\ \hline x_4^2 \\ \hline \end{array} + \mathbf{D} \vec{u}$$

RYСУNEK 31. DEKOMPOZYCJA DWUPOZIOMOWA

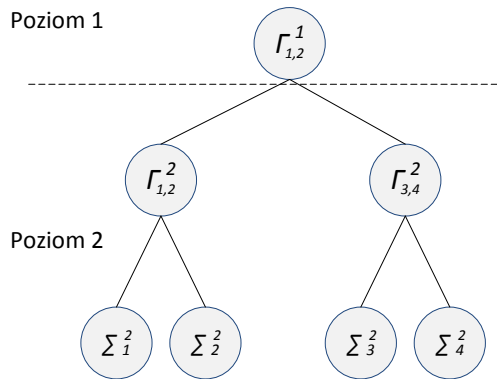
gdzie interfejsy między wydzielonymi podsystemami oznaczono następująco:



RYСУNEK 32. PRZYJĘTA NOTACJA OZNACZANIA INTERFEJSÓW

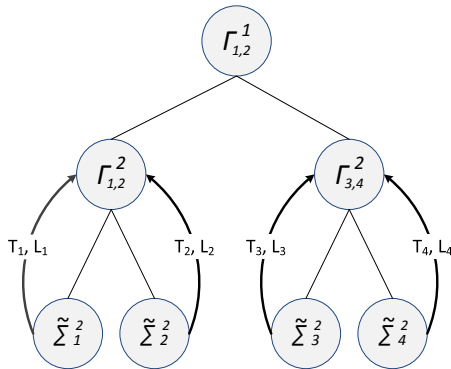
Dekompozycja wielopoziomowa prowadzi do wyodrębnienia kolejnego poziomu podziału macierzy stanu modelu na mniejsze bloki stanowiące modele drugiego poziomu wraz z interfejsami je łączącymi oraz interfejsami występującymi na wyższych poziomach.

Dla uproszczenia, kolejne rozważania będą prowadzone dla przypadku dekompozycji dwupoziomowej, gdzie na każdym poziomie wyróżniono dwa podsystemy. Rozważania te w łatwy sposób można przenieść na dowolną liczbę poziomów, ze względu na fakt, że interesujące z punktu widzenia redukcji modelu są tylko modele wyznaczone na najniższym poziomie dekompozycji. Wyznaczone macierze transformacji dla modeli stanowiących najniższy poziom grafu wykorzystywane są stopniowo w celu modyfikacji kolejnych interfejsów. Dla dekompozycji dwupoziomowej graf dekompozycji został przedstawiony na rysunku 33.

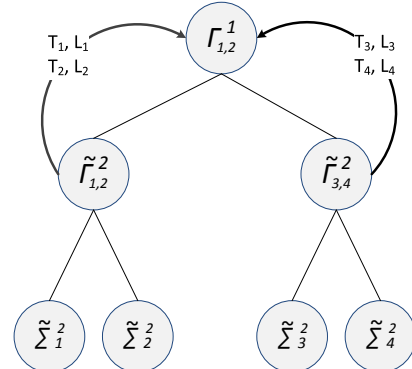


RYSUNEK 33. GRAF DEKOMPOZYCJI DWUPOZIOMEJ

Proces redukcji w ogólnej postaci może zostać zatem wyrażony jako modyfikacja modeli drugiego poziomu oraz kolejnych interfejsów z wykorzystaniem macierzy transformacji wyznaczonych podczas redukcji modeli drugiego poziomu (rys.34, 35).



RYSUNEK 34. MODYFIKACJA INTERFEJSÓW DRUGIEGO POZIOMU



RYSUNEK 35. MODYFIKACJA INTERFEJSU PIERWSZEGO POZIOMU

W celu zastosowania macierzy transformacji dla interfejsów poziomów wyższych konieczna jest ich konkatencja. Dla rozważanego przypadku dekompozycji dwupoziomowej otrzymuje się następujące macierze transformacji pierwszego poziomu (rys.36).

$$T_1^1 = \begin{bmatrix} T_1^2 & 0 \\ 0 & T_2^2 \end{bmatrix} \quad L_1^1 = \begin{bmatrix} L_1^2 & 0 \\ 0 & L_2^2 \end{bmatrix} \quad T_2^1 = \begin{bmatrix} T_3^2 & 0 \\ 0 & T_4^2 \end{bmatrix} \quad L_2^1 = \begin{bmatrix} L_3^2 & 0 \\ 0 & L_4^2 \end{bmatrix}$$

RYSUNEK 36. KONKATENACJA MACIERZY TRANSFORMACJI

Bloki interfejsów zawarte w macierzach E i/lub A zostają zmodyfikowane następująco:

$$\tilde{\Gamma}_{2,1}^{E^2} = T_1^2 \Gamma_{2,1}^{E^2} L_2^2, \quad \tilde{\Gamma}_{1,2}^{E^2} = T_2^2 \Gamma_{1,2}^{E^2} L_1^2, \quad (89)$$

$$\tilde{\Gamma}_{4,3}^{E^2} = T_3^2 \Gamma_{4,3}^{E^2} L_4^2, \quad \tilde{\Gamma}_{3,4}^{E^2} = T_4^2 \Gamma_{3,4}^{E^2} L_3^2, \quad (90)$$

$$\tilde{\Gamma}_{2,1}^{E^1} = T_1^1 \Gamma_{2,1}^{E^1} L_2^1, \quad \tilde{\Gamma}_{1,2}^{E^1} = T_2^1 \Gamma_{1,2}^{E^1} L_1^1, \quad (91)$$

$$\tilde{\Gamma}_{2,1}^{A^2} = T_1^2 \Gamma_{2,1}^{A^2} L_2^2, \quad \tilde{\Gamma}_{1,2}^{A^2} = T_2^2 \Gamma_{1,2}^{A^2} L_1^2, \quad (92)$$

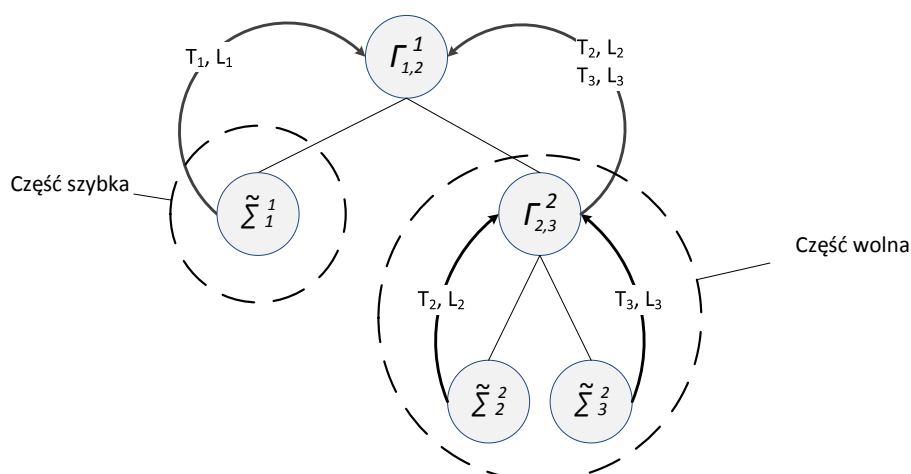
$$\tilde{\Gamma}_{4,3}^{A^2} = T_3^2 \Gamma_{4,3}^{A^2} L_4^2, \quad \tilde{\Gamma}_{3,4}^{A^2} = T_4^2 \Gamma_{3,4}^{A^2} L_3^2, \quad (93)$$

$$\tilde{\Gamma}_{2,1}^{A^1} = T_1^1 \Gamma_{2,1}^{A^1} L_2^1, \quad \tilde{\Gamma}_{1,2}^{A^1} = T_2^1 \Gamma_{1,2}^{A^1} L_1^1, \quad (94)$$

W kolejnych punktach w skrócie przedstawiono wybrane dekompozycje dwupoziomowe.

4.2.1. Dekompozycja slow-fast na pierwszym poziomie dekompozycji

Ze względu na fakt, że rozważane modele rur ekranowych parownika kotła energetycznego BP-1150 posiadają znacznie mniejszy rozmiar części szybkiej w porównaniu z częścią wolną, dalsza dekompozycja części szybkiej wspomnianych modeli jest zbędna. Schemat realizacji redukcji z wykorzystaniem dekompozycji dwupoziomowej, gdzie dekompozycja slow-fast wykonywana jest na pierwszym poziomie dekompozycji przedstawiono na rysunku 37:



RYСУNEK 37. GRAF DEKOMPOZYCJI SLOW-FAST NA PIERWSZYM POZIOMIE DEKOMPOZYCJI

Modyfikacja interfejsu pierwszego poziomu w rozważanym przypadku wyrażona jest zależnościami:

$$\tilde{\Gamma}^{E^1}_{2,1} = T_1^2 \Gamma^{E^1}_{2,1} L_1^1, \quad \tilde{\Gamma}^{E^1}_{1,2} = T_2^1 \Gamma^{E^1}_{1,2} L_1^2, \quad (95)$$

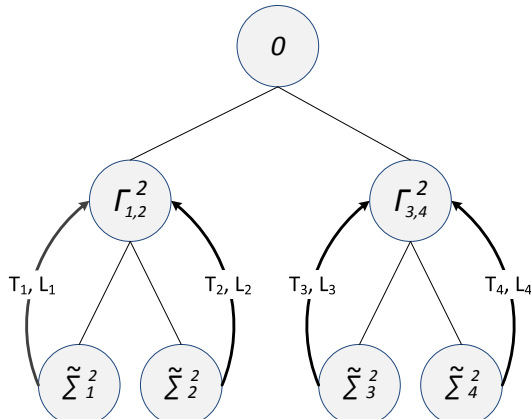
$$\tilde{\Gamma}^{A^1}_{2,1} = T_1^2 \Gamma^{A^1}_{2,1} L_1^1, \quad \tilde{\Gamma}^{A^1}_{1,2} = T_2^1 \Gamma^{A^1}_{1,2} L_1^2, \quad (96)$$

gdzie macierze T_1^2 oraz L_1^2 wynikają bezpośrednio z procesu redukcji części szybkiej.

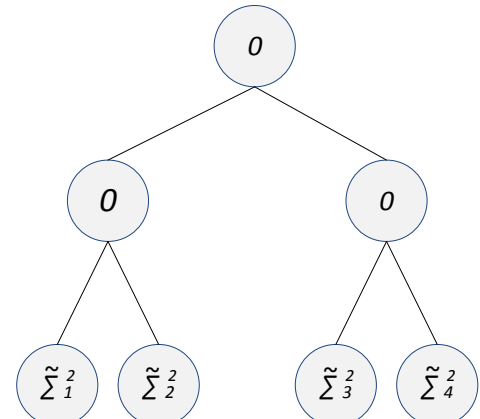
4.2.2. Dekompozycja Schur-Sylvester

Dekompozycja Schur-Sylvester eliminuje połączenia występujące między wyróżnionymi dwoma podsystemami. Realizacja tej metody na pierwszym poziomie dekompozycji prowadzi do uzyskania grafu dekompozycji przedstawionego na rysunku 38.

Poprzez oznaczenie 0 przedstawiono brak powiązania modeli wyznaczonych na pierwszym poziomie dekompozycji (brak wpływu zmiennych stanu jednego podsystemu na zmienne stanu podsystemu drugiego). Realizacja dekompozycji Schur-Sylvester na wszystkich poziomach eliminuje konieczność uwzględniania interfejsów w modelu scalonym (rys. 39).

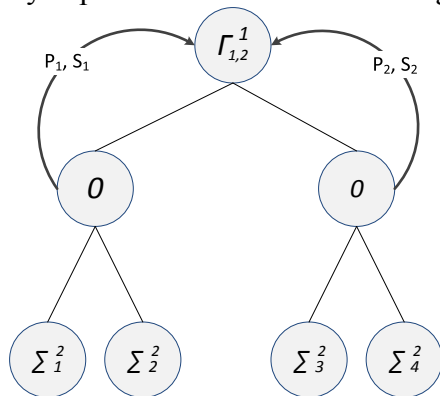


RYSUNEK 38. GRAF DEKOMPOZYCJI SCHUR-SYLVESTER NA PIERWSZYM POZIOMIE DEKOMPOZYCJI

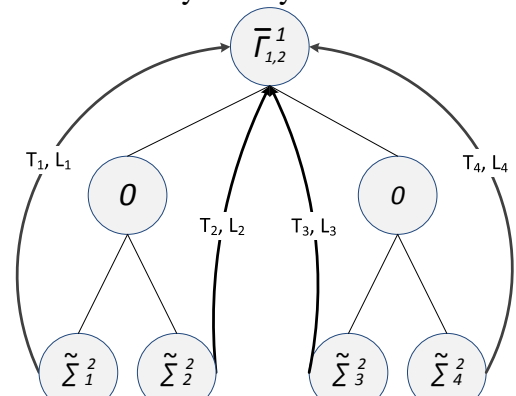


RYSUNEK 39. GRAF DEKOMPOZYCJI SCHUR-SYLVESTER NA OBYDWU POZIOMACH DEKOMPOZYCJI

W przypadku, gdy dekompozycja Schur-Sylvester zostaje zastosowana na drugim poziomie, podczas gdy pierwszy poziom stanowi dekompozycja innego typu (nie eliminująca interfejsów między wydzielanymi podsystemami) konieczne jest przekształcenie macierzy interfejsów macierzami wykorzystanymi w celu realizacji dekompozycji. Schemat realizowanych przekształceń zilustrowano grafem zamieszczonym na rysunku 40.



RYSUNEK 40. GRAF DEKOMPOZYCJI SCHUR-SYLVESTER NA DRUGIM POZIOMIE DEKOMPOZYCJI (przekształcenie interfejsu pierwszego poziomu macierzami przekształceń dekompozycji)



RYSUNEK 41. GRAF DEKOMPOZYCJI SCHUR-SYLVESTER NA DRUGIM POZIOMIE DEKOMPOZYCJI (przekształcenie interfejsu pierwszego poziomu macierzami transformacji redukcji)

Macierze P_1, S_1, P_2, S_2 - macierze przekształceń realizujące dekompozycję Schur-Sylvester na poziomie drugim, wyrażone wzorami:

$$P_1 = V_1^* - X V_2^*, \quad P_2 = V_2^*, \quad S_1 = V_1, \quad S_2 = V_1 X + V_2, \quad (97)$$

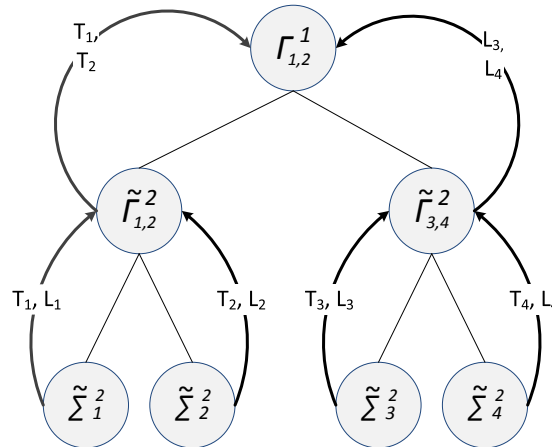
natomiast macierze V_1, V_2 uzyskiwane są na podstawie ortogonalnej macierzy U rzeczywistej dekompozycji Schur'a:

$$U = [V_1 \quad V_2], \quad (98)$$

Macierz X stanowi rozwiązanie równania Sylwestera. W kolejnym kroku interfejs poziomu pierwszego przekształcany jest z wykorzystaniem macierzy $T_1, T_2, T_3, T_4, L_1, L_2, L_3, L_4$, uzyskanych w wyniku redukcji podsystemów drugiego poziomu (rys. 41):

4.2.3. Dekompozycja Schur'a

Dekompozycja Schur'a eliminuje jeden interfejs łączący modele na danym poziomie dekompozycji. W przypadku realizacji dekompozycji Schur'a na pierwszym poziomie dekompozycji uzyskany zostaje graf zamieszczony na rysunku 42.



RYSUNEK 42. GRAF DEKOMPOZYCJI SCHUR'A NA PIERWSZYM POZIOMIE DEKOMPOZYCJI (przekształcanie interfejsów macierzami transformacji redukcji)

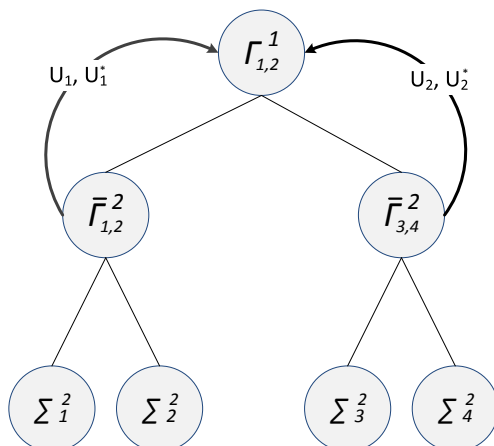
W modelu pierwszego poziomu występuje tylko interfejs łączący model drugi z modelem pierwszym w postaci bloku macierzy A_{12} , modyfikacja interfejsu może zostać zapisana następująco:

$$\tilde{\Gamma}_{2,1}^{A'} = T_1^1 A_{12} L_2^1 \quad (99)$$

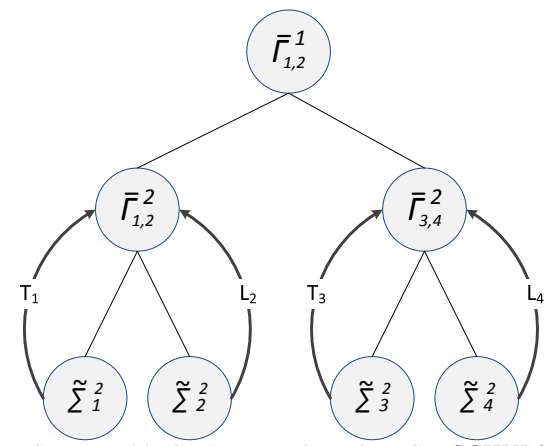
gdzie:

$$T_1^1 = \begin{bmatrix} T_1^2 & 0 \\ 0 & T_2^2 \end{bmatrix}, L_2^1 = \begin{bmatrix} L_3^2 & 0 \\ 0 & L_4^2 \end{bmatrix} \quad (100)$$

Realizacja dekompozycji Schur'a na drugim poziomie związana jest z koniecznością wykonania dwuetapowych przekształceń kolejnych interfejsów. W pierwszym kroku, podobnie jak dla dekompozycji Schur-Sylvester, należy zastosować macierze transformacji wyznaczone w procesie dekompozycji (rys. 43).

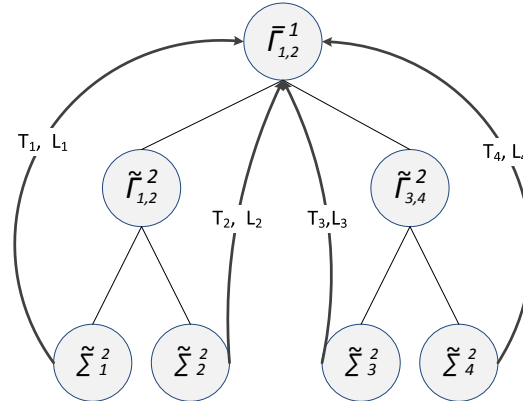


RYSUNEK 43. GRAF DEKOMPOZYCJI SCHUR'A NA DRUGIM POZIOMIE DEKOMPOZYCJI (przekształcanie interfejsu pierwszego poziomu macierzami przekształceń dekompozycji)



RYSUNEK 44. GRAF DEKOMPOZYCJI SCHUR'A NA DRUGIM POZIOMIE DEKOMPOZYCJI (przekształcanie interfejsów drugiego poziomu macierzami transformacji redukcji)

Macierze U stanowią ortogonalne macierze dekompozycji Schur'a. Następnie wykonywane jest przekształcenie interfejsów drugiego poziomu z wykorzystaniem wyznaczonych w procesie redukcji macierzy transformacji (rys. 44). W ostatnim kroku wyznaczone w procesie redukcji macierze transformacji wykorzystywane są do modyfikacji interfejsu pierwszego poziomu (rys. 45).



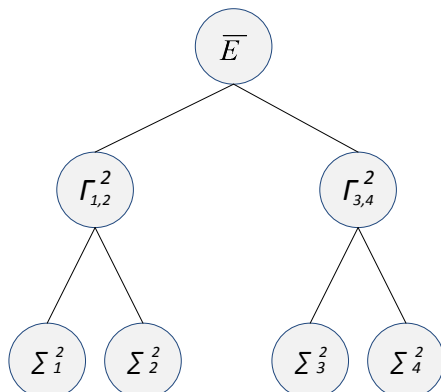
RYSUNEK 45. GRAF DEKOMPOZYCJI SCHUR'A NA DRUGIM POZIOMIE DEKOMPOZYCJI (przekształcenie interfejsu pierwszego poziomu macierzami transformacji redukcji)

4.2.4. Dekompozycja Schur Complement

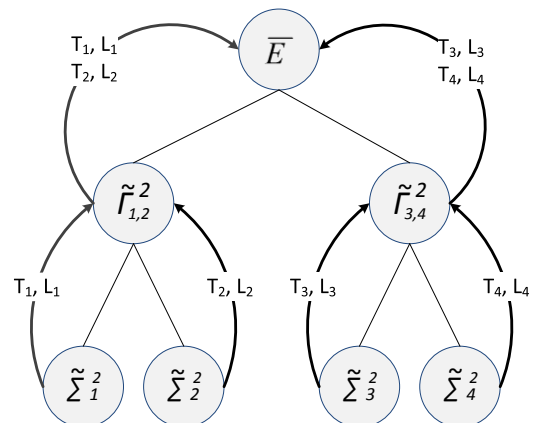
Dekompozycja Schur Complement eliminuje interfejsy realizowane jako bloki macierzy stanu A modelu: A_{12} , A_{21} , jednak kosztem tego zostają wprowadzone niezerowe interfejsy zawarte w macierzy E : \bar{E}_{12} i \bar{E}_{21} , które przenoszą relacje występujące między podsystemami ze zmiennych stanu do pochodnych zmiennych stanu. Realizacja dekompozycji Schur complement na pierwszym poziomie dekompozycji prowadzi do wydzielenia dwóch modeli, określonych zależnościami (69). Interfejs pomiędzy podsystemami zostaje przeniesiony do macierzy \bar{E} :

$$\bar{E} = \begin{bmatrix} I & \bar{E}_{12} \\ \bar{E}_{21} & I \end{bmatrix} \quad (101)$$

zgodnie ze schematem przedstawionym na rysunku 46.

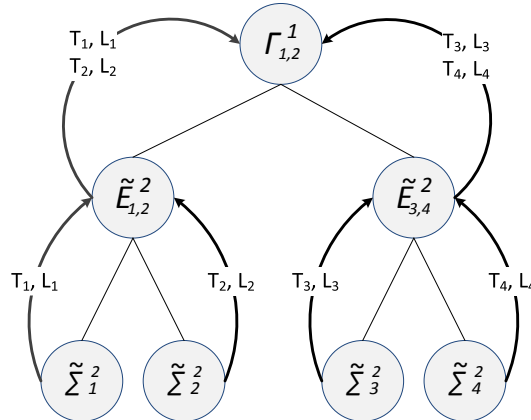


RYSUNEK 46. GRAF DEKOMPOZYCJI SCHUR COMPLEMENT NA PIERWSZYM POZIOMIE DEKOMPOZYCJI



RYSUNEK 47. GRAF DEKOMPOZYCJI SCHUR COMPLEMENT NA PIERWSZYM POZIOMIE DEKOMPOZYCJI (modyfikacja interfejsu poziomu pierwszego macierzami transformacji redukcji)

Macierz \bar{E} nie jest brana pod uwagę w procesie redukcji modeli na poziomie drugim. Wyznaczone w procesie redukcji macierze transformacji wykorzystywane są do modyfikacji interfejsu zawartego w macierzy \bar{E} , w wyniku czego uzyskiwana jest macierz \tilde{E} (rys. 47). Macierz \tilde{E} następnie jest eliminowana z równań stanu poprzez lewostronne mnożenie macierzy A i B modelu przez jej odwrotność. Realizacja dekompozycji Schur Complement na drugim poziomie może zostać przedstawiona za pomocą grafu zamieszczonego na rysunku 48.



RYSUNEK 48. GRAF DEKOMPOZYCJI SCHUR COMPLEMENT NA DRUGIM POZIOMIE DEKOMPOZYCJI

W modelu wynikowym macierze $\tilde{E}_{1,2}^2$ i $\tilde{E}_{3,4}^2$ są eliminowane poprzez mnożenie lewostronne macierzy stanu A i B przez macierze $(\tilde{E}_{1,2}^2)^{-1}$ i $(\tilde{E}_{3,4}^2)^{-1}$. W przypadku realizacji dwupoziomowej dekompozycji Schur Complement zależności zostały wyprowadzone w punkcie 4.2.5.

4.2.5. Dwupoziomowa dekompozycja Schur complement

W przypadku dwupoziomowej dekompozycji opartej na dopełnieniu Schur'a, gdzie dekompozycja skutkuje wydzieleniem czterech podsystemów, równania definiujące zależności pomiędzy poszczególnymi podsystemami mogą zostać wyrażone w następującej formie (rys. 49).

$$\begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u$$

$$\begin{bmatrix} E_{31} & E_{32} \\ E_{41} & E_{42} \end{bmatrix} \begin{bmatrix} \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} A_{31} & A_{32} \\ A_{41} & A_{42} \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} B_3 \\ B_4 \end{bmatrix} u$$

$$y = \begin{bmatrix} C_1 & C_2 & C_3 & C_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

RYSUNEK 49. WYDZIELENIE BŁOKÓW REPREZENTUJĄCYCH CZTERY PODSYSTEMY W RÓWNIANIACH STANU MODELU

Na czerwono zaznaczono macierze należące do podsystemu 1, na czarno do podsystemu 2, natomiast kolorem niebieskim interfejsy pomiędzy podsystemami. W pierwszej kolejności wykonywana jest dekompozycja modelu na dwie części, z pominięciem interfejsów między podsystemami:

$$\begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \\ \dot{\hat{x}}_3 \\ \dot{\hat{x}}_4 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & \emptyset & \emptyset \\ A_{21} & A_{22} & \emptyset & \emptyset \\ \emptyset & \emptyset & \hat{A}_{33} & \hat{A}_{34} \\ \emptyset & \emptyset & \hat{A}_{43} & \hat{A}_{44} \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \end{bmatrix} - \begin{bmatrix} \emptyset & \emptyset \\ \hat{E}_{31} & \hat{E}_{32} \\ \hat{E}_{41} & \hat{E}_{42} \\ \emptyset & \emptyset \end{bmatrix} \begin{bmatrix} \bar{E}_{13} & \bar{E}_{14} \\ \bar{E}_{23} & \bar{E}_{24} \\ \emptyset & \emptyset \\ \emptyset & \emptyset \end{bmatrix} \begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \\ \dot{\hat{x}}_3 \\ \dot{\hat{x}}_4 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \\ \hat{B}_3 \\ \hat{B}_4 \end{bmatrix} u \quad (102)$$

$$y = \begin{bmatrix} C_1 & C_2 \\ \hat{C}_3 & \hat{C}_4 \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \end{bmatrix} \quad (103)$$

W kolejnym kroku następuje dekompozycja każdego podsystemu na dwa podsystemy drugiego poziomu. W wyniku przekształceń zostają otrzymane równania (104) i (105).

$$\begin{bmatrix} I & \bar{E}_{12} & \bar{E}_{13} & \bar{E}_{14} \\ \bar{E}_{22}^{-1} \bar{E}_{21} & I & \bar{E}_{22}^{-1} \bar{E}_{23} & \bar{E}_{22}^{-1} \bar{E}_{24} \\ \hat{E}_{31} & \hat{E}_{32} & I & \bar{E}_{34} \\ \bar{E}_{44}^{-1} \hat{E}_{41} & \bar{E}_{44}^{-1} \hat{E}_{42} & \bar{E}_{44}^{-1} \bar{E}_{43} & I \end{bmatrix} \begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \\ \dot{\hat{x}}_3 \\ \dot{\hat{x}}_4 \end{bmatrix} = \begin{bmatrix} A_{11} & \emptyset & \emptyset & \emptyset \\ \emptyset & \bar{E}_{22}^{-1} \bar{A}_{22}^{-1} & \emptyset & \emptyset \\ \emptyset & \emptyset & \hat{A}_{33} & \emptyset \\ \emptyset & \emptyset & \emptyset & \bar{E}_{44}^{-1} \hat{A}_{44} \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \end{bmatrix} + \begin{bmatrix} B_1 \\ \bar{E}_{22}^{-1} \bar{B}_2 \\ \hat{B}_3 \\ \bar{E}_{44}^{-1} \hat{B}_4 \end{bmatrix} u \quad (104)$$

$$y = \begin{bmatrix} C_1 & \bar{C}_2 & \bar{C}_3 & \bar{C}_4 \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \end{bmatrix} \quad (105)$$

gdzie:

$$\bar{E}_{12} = -A_{11}^{-1} A_{12}, \quad \bar{E}_{13}, \quad \begin{bmatrix} \bar{E}_{13} & \bar{E}_{14} \\ \bar{E}_{23} & \bar{E}_{24} \end{bmatrix} = -\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{-1} \begin{bmatrix} A_{13} & A_{14} \\ A_{23} & A_{24} \end{bmatrix}, \quad \bar{E}_{14} = \bar{E}_{13} \hat{A}_{33}^{-1} \hat{A}_{34} + \bar{E}_{14}, \quad (106)$$

$$\begin{bmatrix} \hat{A}_{33} & \hat{A}_{34} \\ \hat{A}_{43} & \hat{A}_{44} \end{bmatrix} = \begin{bmatrix} \bar{E}_{33} & \bar{E}_{34} \\ \bar{E}_{43} & \bar{E}_{44} \end{bmatrix}^{-1} \begin{bmatrix} \bar{A}_{33} & \bar{A}_{34} \\ \bar{A}_{43} & \bar{A}_{44} \end{bmatrix}, \quad (107)$$

$$\begin{bmatrix} \bar{E}_{33} & \bar{E}_{34} \\ \bar{E}_{43} & \bar{E}_{44} \end{bmatrix} = I + \begin{bmatrix} A_{31} & A_{32} \\ A_{41} & A_{42} \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{-1} \begin{bmatrix} A_{13} & A_{14} \\ A_{23} & A_{24} \end{bmatrix}, \quad (108)$$

$$\begin{bmatrix} \bar{A}_{33} & \bar{A}_{34} \\ \bar{A}_{43} & \bar{A}_{44} \end{bmatrix} = \begin{bmatrix} A_{33} & A_{34} \\ A_{43} & A_{44} \end{bmatrix} - \begin{bmatrix} A_{31} & A_{32} \\ A_{41} & A_{42} \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{-1} \begin{bmatrix} A_{13} & A_{14} \\ A_{23} & A_{24} \end{bmatrix}, \quad \bar{E}_{22} = I + A_{21} A_{11}^{-1} A_{12}, \quad (109)$$

$$\bar{E}_{23} = -A_{21} A_{11}^{-1} \bar{E}_{13} + E_{23}, \quad \bar{E}_{24} = A_{21} A_{11}^{-1} \bar{E}_{13} \hat{A}_{33}^{-1} \hat{A}_{34} - \bar{E}_{23} \hat{A}_{33}^{-1} \hat{A}_{34} - A_{21} A_{11}^{-1} \bar{E}_{14} + \bar{E}_{24}, \quad (110)$$

$$\bar{A}_{22} = A_{22} - A_{21} A_{11}^{-1} A_{12}, \quad \bar{B}_2 = B_2 - A_{21} A_{11}^{-1} B_1, \quad \bar{C}_2 = C_2 - C_1 A_{11}^{-1} A_{12}, \quad (111)$$

$$\begin{bmatrix} \hat{E}_{31} & \hat{E}_{32} \\ \hat{E}_{41} & \hat{E}_{42} \end{bmatrix} = \begin{bmatrix} \bar{E}_{33} & \bar{E}_{34} \\ \bar{E}_{43} & \bar{E}_{44} \end{bmatrix}^{-1} \begin{bmatrix} \bar{E}_{31} & \bar{E}_{32} \\ \bar{E}_{41} & \bar{E}_{42} \end{bmatrix}, \quad \hat{E}_{32} = \hat{E}_{32} - \hat{E}_{31} A_{11}^{-1} A_{12}, \quad (112)$$

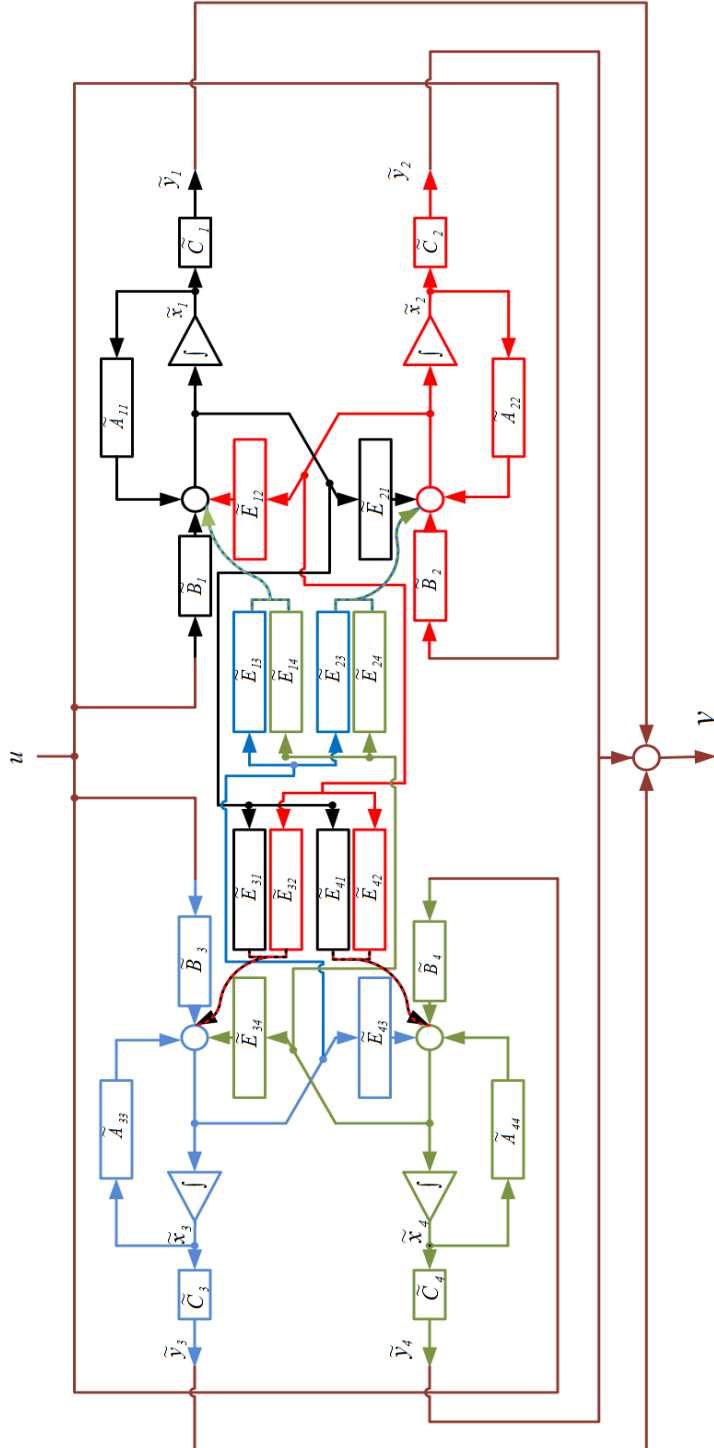
$$\begin{bmatrix} \hat{B}_3 \\ \hat{B}_4 \end{bmatrix} = \begin{bmatrix} \bar{E}_{33} & \bar{E}_{34} \\ \bar{E}_{43} & \bar{E}_{44} \end{bmatrix}^{-1} \begin{bmatrix} \bar{B}_3 \\ \bar{B}_4 \end{bmatrix}, \quad \begin{bmatrix} \bar{B}_3 \\ \bar{B}_4 \end{bmatrix} = \begin{bmatrix} B_3 \\ B_4 \end{bmatrix} - \begin{bmatrix} A_{31} & A_{32} \\ A_{41} & A_{42} \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{-1} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad (113)$$

$$\begin{bmatrix} \bar{C}_3 & \bar{C}_4 \end{bmatrix} = \begin{bmatrix} C_3 & C_4 \end{bmatrix} - \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{-1} \begin{bmatrix} A_{13} & A_{14} \\ A_{23} & A_{24} \end{bmatrix}, \quad (114)$$

$$\hat{E}_{41} = -\hat{A}_{43}\hat{A}_{33}^{-1}\hat{E}_{31} + \hat{E}_{41}, \quad \hat{E}_{42} = \hat{A}_{43}A_{33}^{-1}\hat{E}_{32}A_{11}^{-1}A_{12} - \hat{E}_{41}A_{11}^{-1}A_{12} - \hat{A}_{43}\hat{A}_{33}^{-1}\hat{E}_{32} + \hat{E}_{42}, \quad (115)$$

$$\hat{A}_{44} = \hat{A}_{44} - \hat{A}_{43}\hat{A}_{33}^{-1}\hat{A}_{34}, \quad \hat{B}_4 = \hat{B}_4 - \hat{A}_{43}\hat{A}_{33}^{-1}\hat{B}_3, \quad \bar{C}_4 = \bar{C}_4 - \bar{C}_3\hat{A}_{33}^{-1}\hat{A}_{34}, \quad (116)$$

Schemat łączenia podsystemów po wykonaniu operacji redukcji został przedstawiony na rysunku 50.

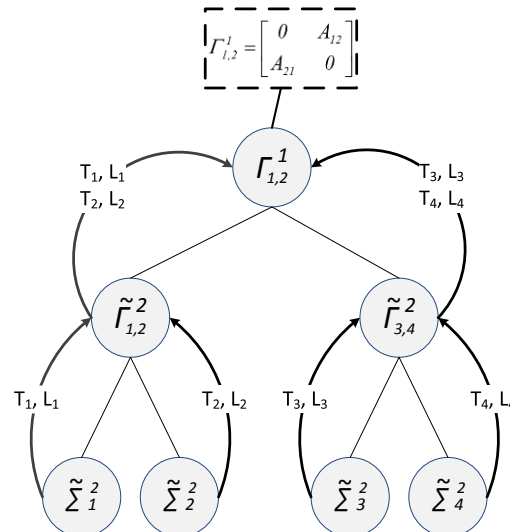


$$\begin{aligned} \tilde{E}_{12} &= T_1\bar{E}_{12}L_2, \\ \tilde{E}_{13} &= T_1\bar{E}_{13}L_3, \\ \tilde{E}_{14} &= T_1\bar{E}_{14}L_4, \\ \tilde{E}_{21} &= T_2\bar{E}_{22}^{-1}E_{21}L_1, \\ \tilde{E}_{23} &= T_2\bar{E}_{22}^{-1}\bar{E}_{23}L_3, \\ \tilde{E}_{24} &= T_2\bar{E}_{22}^{-1}\bar{E}_{24}L_4, \\ \tilde{E}_{31} &= T_3\hat{E}_{31}L_1, \\ \tilde{E}_{32} &= T_3\hat{E}_{32}L_2, \\ \tilde{E}_{34} &= T_3\bar{E}_{34}L_4, \\ \tilde{E}_{41} &= T_4\bar{E}_{44}^{-1}\hat{E}_{41}L_1, \\ \tilde{E}_{42} &= T_4\bar{E}_{44}^{-1}\hat{E}_{42}L_2, \\ \tilde{E}_{43} &= T_4\bar{E}_{44}^{-1}\bar{E}_{43}L_3, \\ \tilde{A}_{11} &= T_1A_{11}L_1, \\ \tilde{A}_{22} &= T_2\bar{E}_{22}^{-1}\bar{A}_{22}^{-1}L_2, \\ \tilde{A}_{33} &= T_3\hat{A}_{33}L_3, \\ \tilde{A}_{44} &= T_4\bar{E}_{44}^{-1}\hat{A}_{44}L_4, \\ \tilde{B}_1 &= T_1B_1, \\ \tilde{B}_2 &= T_2\bar{E}_{22}^{-1}\bar{B}_2, \\ \tilde{B}_3 &= T_3\hat{B}_3, \\ \tilde{C}_1 &= C_1L_1, \\ \tilde{C}_2 &= \bar{C}_2L_2, \\ \tilde{C}_3 &= \bar{C}_3L_3, \\ \tilde{C}_4 &= \bar{C}_4L_4. \end{aligned}$$

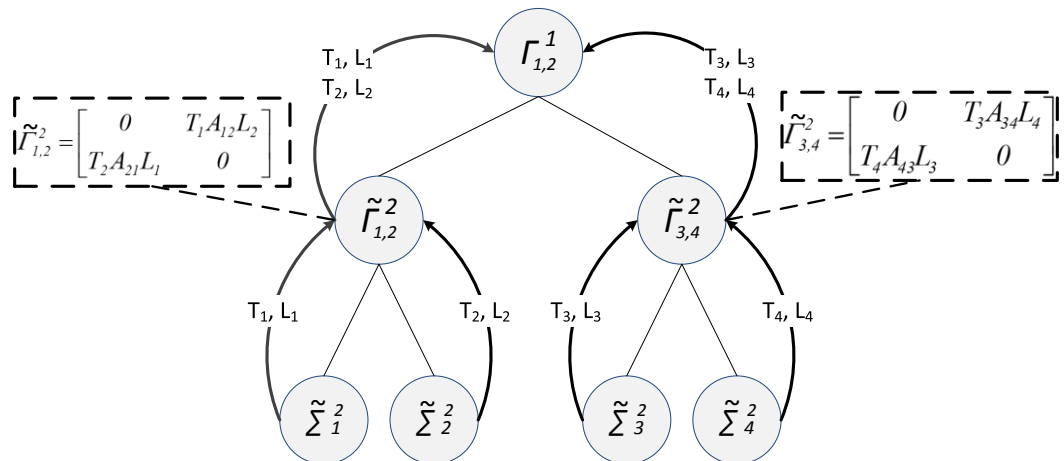
RYСУNEK 50. ŁĄCZENIE ZREDUKOWANYCH MACIERZY Z WYKORZYSTANIEM DWUPOZIOMOWEJ DEKOMPOZYCJI OPARTEJ NA DOPEŁNIENIU SCHUR'A

4.2.6. Dekompozycja strukturalna i strukturalna uproszczona

Dekompozycja strukturalna i strukturalna uproszczona bazują na realizacji interfejsów pomiędzy podsystemami w postaci macierzy A_{12} , A_{21} . Realizacja obu dekompozycji dla potrzeb redukcji jest podobna, jednak w przypadku dekompozycji strukturalnej interfejsy pomiędzy modelami podsystemów uwzględniane są w modelach zdekomponowanych jako dodatkowe kolumny macierzy stanu B lub dodatkowe wiersze macierzy stanu C. Graf redukcji dla dekompozycji strukturalnej realizowanej na poziomie pierwszym i drugim został przedstawiony na rysunkach 51 i 52.



RYSUNEK 51. GRAF DEKOMPOZYCJI STRUKUTRALNEJ NA PIERWSZYM POZIOMIE DEKOMPOZYCJI



RYSUNEK 52. GRAF DEKOMPOZYCJI STRUKUTRALNEJ NA DRUGIM POZIOMIE DEKOMPOZYCJI

4.2.7. Zmodyfikowana dekompozycja Schur complement

Z punktu widzenia realizacji, redukcja z wykorzystaniem zmodyfikowanej dekompozycji Schur complement wygląda analogicznie jak dekompozycja strukturalna. Interfejs pomiędzy wydzielonymi dwoma podsystemami realizowany jest poprzez bloki macierzy A:

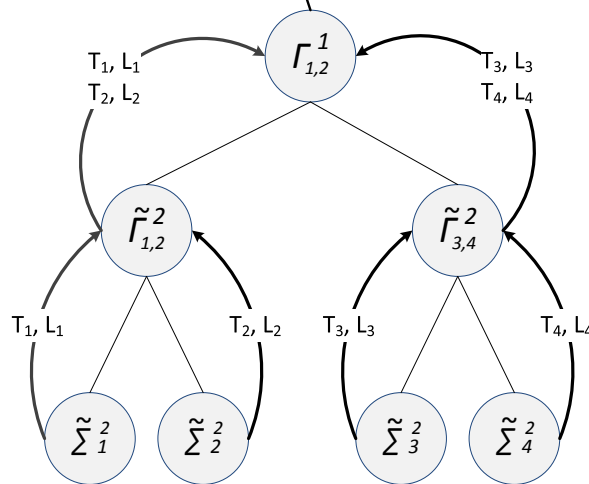
$$\Gamma = \begin{bmatrix} 0 & -(I - \bar{E}_{12} \bar{E}_{22}^{-1} \bar{E}_{21})^{-1} \bar{E}_{12} \bar{E}_{22}^{-1} \bar{A}_{22} \\ -(I - \bar{E}_{22}^{-1} \bar{E}_{21} \bar{E}_{12})^{-1} \bar{E}_{22}^{-1} \bar{E}_{21} \bar{A}_{11} & 0 \end{bmatrix} \quad (117)$$

Dla realizacji dekompozycji na pierwszym i drugim poziomie, grafy dekompozycji zostały zamieszczone na rysunkach 53 i 54.

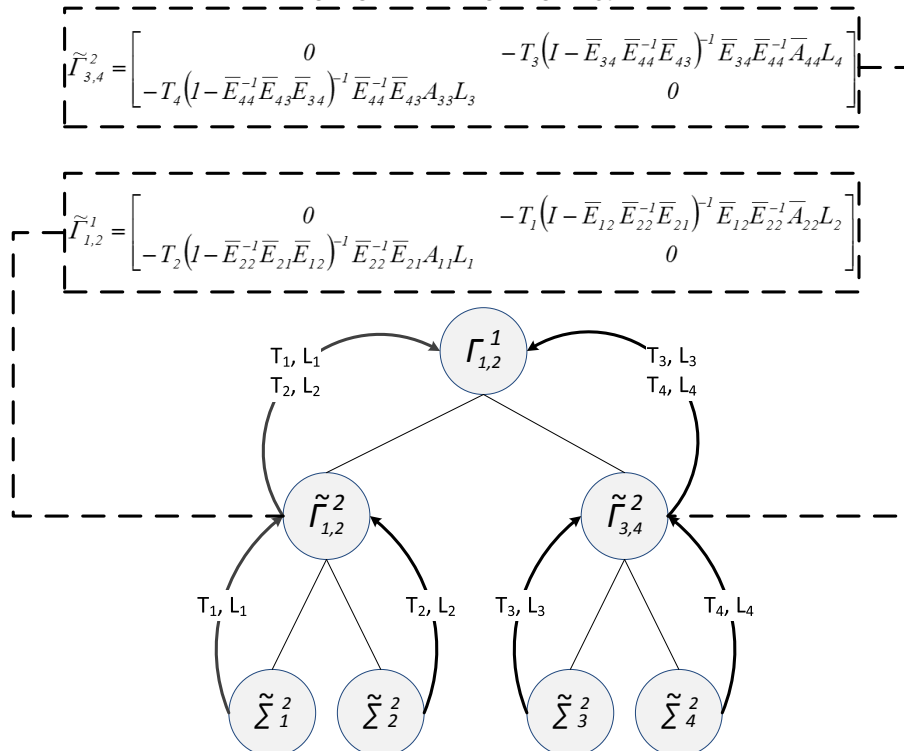
$$\tilde{\Gamma}_{1,2}^1 = \begin{bmatrix} 0 & -T_1(I - \bar{E}_{12} \bar{E}_{22}^{-1} \bar{E}_{21})^{-1} \bar{E}_{12} \bar{E}_{22}^{-1} \bar{A}_{22} L_2 \\ -T_2(I - \bar{E}_{22}^{-1} \bar{E}_{21} \bar{E}_{12})^{-1} \bar{E}_{22}^{-1} \bar{E}_{21} A_{11} L_1 & 0 \end{bmatrix} \quad (118)$$

$$\tilde{\Gamma}_{3,4}^2 = \begin{bmatrix} 0 & -T_3(I - \bar{E}_{34} \bar{E}_{44}^{-1} \bar{E}_{43})^{-1} \bar{E}_{34} \bar{E}_{44}^{-1} \bar{A}_{44} L_4 \\ -T_4(I - \bar{E}_{44}^{-1} \bar{E}_{43} \bar{E}_{34})^{-1} \bar{E}_{44}^{-1} \bar{E}_{43} A_{33} L_3 & 0 \end{bmatrix} \quad (119)$$

$$\Gamma_{1,2}^1 = \begin{bmatrix} 0 & -(I - \bar{E}_{12} \bar{E}_{22}^{-1} \bar{E}_{21})^{-1} \bar{E}_{12} \bar{E}_{22}^{-1} \bar{A}_{22} \\ -(I - \bar{E}_{22}^{-1} \bar{E}_{21} \bar{E}_{12})^{-1} \bar{E}_{22}^{-1} \bar{E}_{21} A_{11} & 0 \end{bmatrix}$$



RYСУNEK 53. GRAF ZMODYFIKOWANEJ DEKOMPOZYCJI SCHUR COMPLEMENT NA PIERWSZYM POZIOMIE DEKOMPOZYCJI



RYСУNEK 54. GRAF ZMODYFIKOWANEJ DEKOMPOZYCJI SCHUR COMPLEMENT NA DRUGIM POZIOMIE DEKOMPOZYCJI

4.3. Uwagi

Biorąc pod uwagę rys. 30, pomijając dekompozycję fizyczną, w pracy przeanalizowano $13 \cdot 6 = 78$ metod dekompozycji dwupoziomowych. Dekompozycja slow-fast jest stosowana wyłącznie na pierwszym poziomie, ponieważ dla analizowanych w dalszych rozdziałach modeli wydziela podsystem szybki o niewielkim rozmiarze, co powoduje, że na drugim poziomie dekomponowany jest wyłącznie podsystem wolny.

Redukcja modeli ma miejsce wyłącznie na drugim poziomie dekompozycji. Model zredukowany systemu uzyskuje się z zastosowaniem interfejsów na pierwszym i drugim poziomie dekompozycji.

Postać interfejsu ma istotny wpływ na przenoszenie błędu redukcji modeli podsystemów na błąd modelu zredukowanego systemu, jak również na stabilność modelu zredukowanego.

Dla wszystkich metod dekompozycji dwupoziomowych opracowano programy do wyznaczania modeli zredukowanych systemu na podstawie zredukowanych modeli podsystemów.

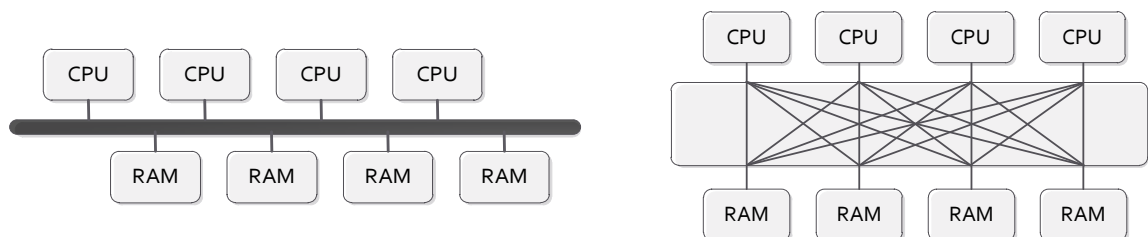
5. Realizacja algorytmów redukcji modeli LTI z zastosowaniem obliczeń równoległych

5.1. Sposoby zrównoleglania procesu obliczeniowego

Proces obliczeniowy realizowany przez systemy komputerowe związany jest z przetwarzaniem strumienia instrukcji operujących na zbiorze danych. Podstawowa klasyfikacja systemów komputerowych, w zależności od sposobu zrównoleglania wykonywanych instrukcji i wektoryzacji danych podlegających przetwarzaniu, została przedstawiona przez Michaela Flynna w 1966 roku. Wyróżnił on cztery podstawowe grupy systemów komputerowych [27]:

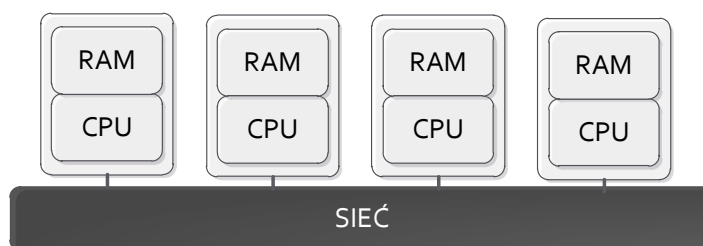
- **SISD** (ang. **Single Instruction Single Data**) – sekwencyjne przetwarzanie danych skalarnych,
- **SIMD** (ang. **Single Instruction Multiple Data**) – sekwencyjne przetwarzanie danych wektorowych,
- **MISD** (ang. **Multiple Instruction Single Data**) – równoległe przetwarzanie danych skalarnych (praktycznie niestosowane),
- **MIMD** (ang. **Multiple Instruction Multiple Data**) – równoległe przetwarzanie wielu zbiorów danych.

Do obliczeń równoległych szczególnie korzystne są komputery MIMD, które cechują się zdolnością przetwarzania wielu instrukcji z wykorzystaniem wielu zbiorów danych jednocześnie. Ten rodzaj przetwarzania danych jest typowy dla procesorów wielordzeniowych, komputerów wieloprocessorowych oraz klastrów komputerowych. Ze względu na sposób dostępu do pamięci, w grupie tej można dokonać dalszego podziału na systemy komputerowe z pamięcią współdzieloną SM-MIMD (Shared Memory – Multiple Instruction Multiple Data), określane mianem wieloprocessorów, oraz z pamięcią rozproszoną DM-MIMD (Distributed Memory – Multiple Instruction Multiple Data), określane mianem wielokomputerów. Procesory systemów SM-MIMD przyłączone są do wspólnej pamięci z wykorzystaniem magistrali danych lub systemu przełącznic krzyżowych (rysunek 55) [125].



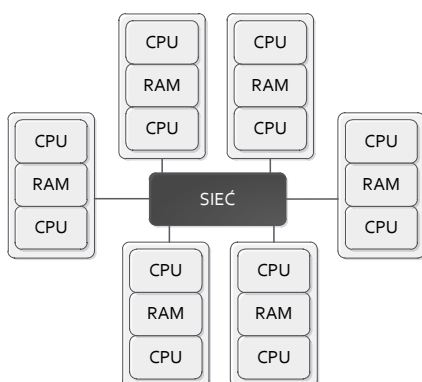
RYСУNEK 55. ARCHITEKTURY SM-MIMD (SYSTEMY Z PAMIĘCIĄ WSPÓLDZIELONĄ)

Ze względu na wysoki koszt realizacji oraz słabe możliwości rozbudowy, systemów SM-MIMD w praktyce częściej wykorzystuje się architekturę DM-MIMD. W architekturze tej każdy procesor dysponuje własną pamięcią lokalną. Połączenie między jednostkami obliczeniowymi realizowane jest przez sieć komputerową [126]. Ponieważ sieć komputerowa zapewnia niższe prędkości przesyłu danych od komunikacji z wykorzystaniem pamięci współdzielonej, rozwiązanie to narzuca konieczność podziału problemu obliczeniowego na części realizowane oddzielnie na poszczególnych jednostkach obliczeniowych z możliwie minimalną komunikacją z pozostałymi węzłami. Schemat obrazujący ideę architektury DM-MIMD został przedstawiony na rysunku 56.

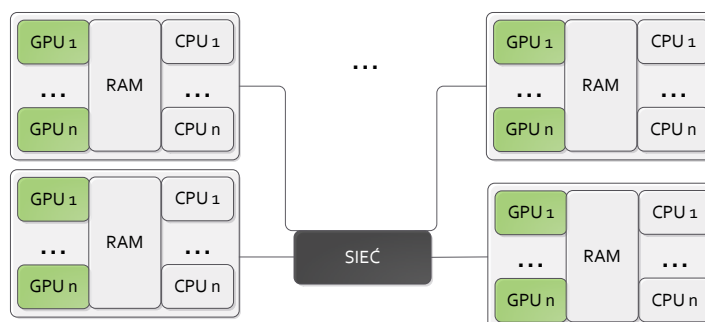


RYSUNEK 56. ARCHITEKTURA DM-MIMD (SYSTEMY Z PAMIĘCIĄ ROZPROSZONĄ)

Najbardziej popularnym rozwiązaniem związanym z wykorzystaniem architektury MIMD jest klastery komputerowy. Klastery komputerowe składają się z komputerów, stanowiących jednostki obliczeniowe, zwanych węzłami (ang. nodes) połączonych wspólną siecią komputerową. W przypadku, gdy węzły klastra stanowią komputery stacjonarne, określa się go mianem klastra stacji roboczych (ang. cluster of workstations). W przypadku współczesnych rozwiązań, typowa architektura klastra stacji roboczych jest połączeniem architektury SM-MIMD oraz DM-MIMD. Stacje robocze wyposażone w procesory wielordzeniowe stanowią systemy z pamięcią współdzieloną. Poszczególne węzły klastra jednak nie mają bezpośredniego dostępu do pamięci pozostałych jednostek, tworząc architekturę DM-MIMD [126]. Schemat przedstawiający architekturę klastra komputerów został przedstawiony na rysunku 57.



RYSUNEK 57. TYPOWA ARCHITEKTURA KLASTROWA



RYSUNEK 58. KLASTER KOMPUTEROWY Z PROCESORAMI GRAFICZNYMI

Coraz częściej stosuje się również architektury, które do procesu obliczeniowego oprócz procesorów klasycznych wykorzystują procesory graficzne (rysunek 58).

Kolejną podgrupę MIMD stanowią systemy realizujące pojedynczy program w celu przetwarzania wielu strumieni danych – SPMD (ang. Single Program Multiple Data). Typowymi przedstawicielami tej grupy są procesory graficzne, w których wszystkie jednostki obliczeniowe wykonują ten sam program, operując na bloku danych [127].

5.2. Architektura Fermi procesorów graficznych firmy NVIDIA

Wykorzystanie mocy obliczeniowej karty graficznej do celów, dla których nie została ona pierwotnie stworzona określane jest mianem obliczeń GPGPU (ang. General-Purpose GPU computing). W niniejszej pracy wykorzystano karty graficzne firmy NVIDIA oparte na architekturze FERMI.

Pojedynczą jednostką odpowiedzialną za wykonywanie obliczeń w procesorze graficznym jest rdzeń CUDA (ang. Compute Unified Device Architecture), określany również skrótami TP (ang. Thread Processor) lub SP (ang. Streaming Processor). Każdy rdzeń CUDA

wyposażony jest w jednostkę arytmetyczno-logiczną dla liczb całkowitych (ALU) oraz jednostkę zmiennoprzecinkową (FPU). W architekturze FERMI każda jednostka zmiennoprzecinkowa obsługuje arytmetykę zgodną ze standardem IEEE 754-2008 dla danych pojedynczej i podwójnej precyzji. Standard IEEE 754-2008 w porównaniu ze standardem IEEE 754-1985, który był używany przez procesory graficzne starszej generacji, wprowadza operację FMA (ang. Fused Multiply-Add), która w jednym kroku wykonuje mnożenie i sumowanie argumentów ($D=A*B+C$), zapewniając większą precyzję niż w przypadku wykonywania obu operacji oddzielnie [128].

Wykonywanie operacji na danych pojedynczej precyzji realizowane jest przez pojedynczy rdzeń CUDA, natomiast operacje wykonujące działania na danych podwójnej precyzji wymagają wykorzystania dwóch rdzeni CUDA. W związku z tym moc obliczeniowa procesora graficznego dla operacji na liczbach podwójnej precyzji jest dwa razy mniejsza (w produktach dostępnych do zastosowań domowych, moc obliczeniowa dla podwójnej precyzji jest jeszcze bardziej ograniczona) [129].

Oprócz jednostek obliczeniowych ALU oraz FPU, w skład pojedynczego rdzenia CUDA wchodzi układy kierujące instrukcje i operandy do właściwej jednostki obliczeniowej oraz kolejka rezultatów. W przeciwieństwie do klasycznych procesorów, rdzenie CUDA nie posiadają własnych rejestrów, pamięci podręcznej czy układów umożliwiających zapis lub odczyt danych z pamięci [128]. Grupa 32 rdzeni CUDA wraz z dodatkowymi jednostkami wchodzi w skład multiprocesora strumieniowego (ang. Streaming Multiprocessor), określanego skrótem SM [130].

Szesnaście jednostek odczytująco/zapisujących (L/S) umożliwia jednoczesne wykonanie szesnastu operacji odwołujących się do pamięci DRAM lub CACHE w celu odczytania operandu instrukcji lub zapisania wyniku działania. Jednostki specjalnego przeznaczenia SFU (ang. Special Function Units) przeznaczone są do wykonywania działań specjalnych, takich jak wyznaczanie wartości funkcji przestępnych, odwrotności czy pierwiastka kwadratowego. Architektura Fermi wspiera sprzętowe zarządzanie wątkami uruchomionymi na procesorze graficznym. Podstawową jednostką alokacji zasobów multiprocesora strumieniowego jest wiązka. Wiazka składa się z 32 wątków. Każdy multiprocesor strumieniowy jest w stanie zarządzać 48 wiązkami, co oznacza, że maksymalna liczba wątków przydzielonych do jednego multiprocesora strumieniowego wynosi 1536 [128]. Ze względu na ograniczoną liczbę rdzeni CUDA, w danej chwili na procesorze strumieniowym mogą wykonywać się maksymalnie 32 wątki (jedna wiązka). Jednostkami odpowiedzialnymi za szeregowanie i przydział instrukcji konkretnej pary wiązek (ze względu na dwie jednostki szeregujące) do zasobów multiprocesora strumieniowego są schedulery wiązek i układy sterujące. Wybierają one po 16 wątków z dwóch wiązek i przydzielają je do jednoczesnego uruchomienia na 16 rdzeniach CUDA, 16 jednostkach zapisująco/odczytujących, lub czterech jednostkach specjalnego przeznaczenia [131]. Odpowiedzialne są one za optymalne wykorzystanie zasobów multiprocesora strumieniowego. W przypadku gdy wykryją, że wykonanie bieżącej operacji w wiązce związane jest z długim czasem opóźnienia (przykładowo odwołanie do pamięci DRAM), przydzielają zasoby multiprocesora strumieniowego instrukcji z innej wiązki. Przełączenie wątków w architekturze Fermi zajmuje zaledwie jeden cykl maszynowy i określane jest mianem zero-overhead thread scheduling (przełączanie wątków z zerowym czasem opóźnienia) [128]. Podejście takie stanowi kluczowe rozwiązanie decydujące o wysokiej efektywności opisywanej architektury zaopatrzonej w niewielką ilość pamięci cache (a co za tym idzie, architektury zaprojektowanej do przetwarzania danych, ukierunkowanej na znaczną liczbę odwołań do pamięci DRAM).

W architekturze Fermi każdy multiprocesor strumieniowy dysponuje pamięcią 64KB o krótkim czasie dostępu, która jest dzielona na dwa bloki odpowiadające pamięci współdzielonej (ang. Shared Memory) oraz pamięci cache L1. Wielkość obu bloków może

zostać skonfigurowana dwojako na bloki o wielkości 48KB dla pamięci współdzielonej oraz 16KB dla pamięci podręcznej L1 lub 16KB dla pamięci współdzielonej oraz 48KB dla pamięci podręcznej L1 [129]. Konkretna konfiguracja jest ustalana na podstawie zachowania wykonywanych wątków. W przypadku, gdy wątki wykonują operacje głównie na danych lokalnych, korzystniejsza jest konfiguracja z większą pamięcią współdzieloną. Dla przypadku, gdy nie można przewidzieć z których obszarów pamięci DRAM wątki będą pobierały dane, korzystniejsza jest konfiguracja z większą pamięcią podręczną L1 [128].

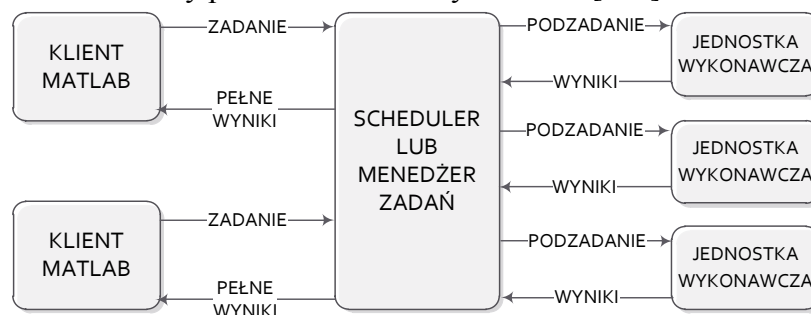
Grupa czterech multiprocessorów strumieniowych tworzy graficzny klaster obliczeniowy GPC (ang. Graphics Processing Cluster). Klastry obliczeniowe wraz z pamięcią L2 (768KB), kontrolerami pamięci DRAM oraz układem zarządzającym wątkami (GigaThred Engine) wchodzi w skład procesora graficznego GPU (ang. Graphics Processing Unit) [132].

5.3. Charakterystyka środowiska obliczeniowego MATLAB

Środowisko obliczeniowe MATLAB stanowi niezwykle rozbudowane narzędzie umożliwiające wykorzystanie tysięcy funkcji matematycznych rozwiązujących rozmaite zagadnienia matematyczne. Współczesne dystrybucje MATLAB'a dostarczają mechanizmy pozwalające użytkownikowi na wykonanie obliczeń równoległych z wykorzystaniem wielu rdzeni procesora (Parallel Computing Toolbox) oraz wielu komputerów połączonych w klaster obliczeniowy (Distributed Computing Engine). Przykładowym mechanizmem wprowadzającym równoległość na poziomie pętli programowych jest instrukcja `parfor`. Instrukcja dzieli poszczególne cykle pętli na grupy, które zostają przydzielone do realizacji na osobnych jednostkach wykonawczych (ang. workers). Wymogiem koniecznym do spełnienia przy wykorzystaniu pętli `parfor` jest niezależność danych z poszczególnych iteracjach od siebie [133].

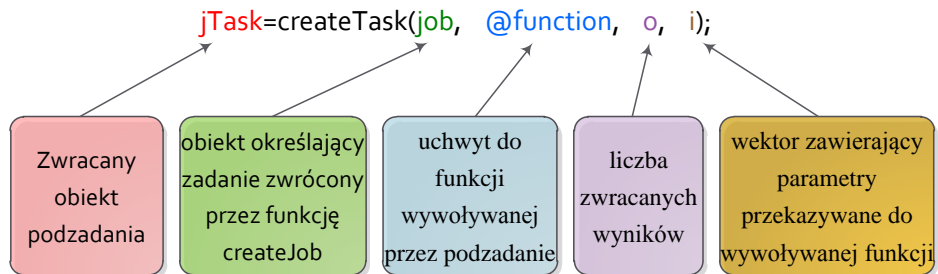
Innym mechanizmem umożliwiającym równoległą realizację bloku instrukcji, z wykorzystaniem wielu jednostek wykonawczych, jest instrukcja `spmd`. Poszczególne instancje programu wykorzystują wartość zmiennej `labindex` w celu identyfikacji. Wewnątrz kodu znajdującego się w bloku `spmd` możliwe jest wykorzystanie instrukcji do komunikacji pomiędzy instancjami programu z wykorzystaniem funkcji `labSend` i `labReceive` [134].

Do podstawowych pojęć związanych z wykorzystaniem klastrowego środowiska MATLAB należy zaliczyć: klient MATLAB (MATLAB client), zadanie (job), podzadanie (task), scheduler zadań (job scheduler) oraz jednostkę wykonawczą (worker). Klient MATLAB stanowi aplikację, która zleca realizację zadań w klastrze. Za zadanie rozumie się zbiór operacji wykonywanych w środowisku MATLAB, które stanowią większą całość. W zadaniu można wyróżnić fragmenty zwane podzadaniami. Scheduler zadań (lub menedżer zadań) jest elementem środowiska, który zajmuje się szeregowaniem zadań i przydzielaniem konkretnych podzadań do wykonania na jednostkach realizujących zadania, zwanych jednostkami wykonawczymi. Schematycznie powiązania wszystkich opisanych elementów środowiska MATLAB zostały przedstawione na rysunku 59 [135].



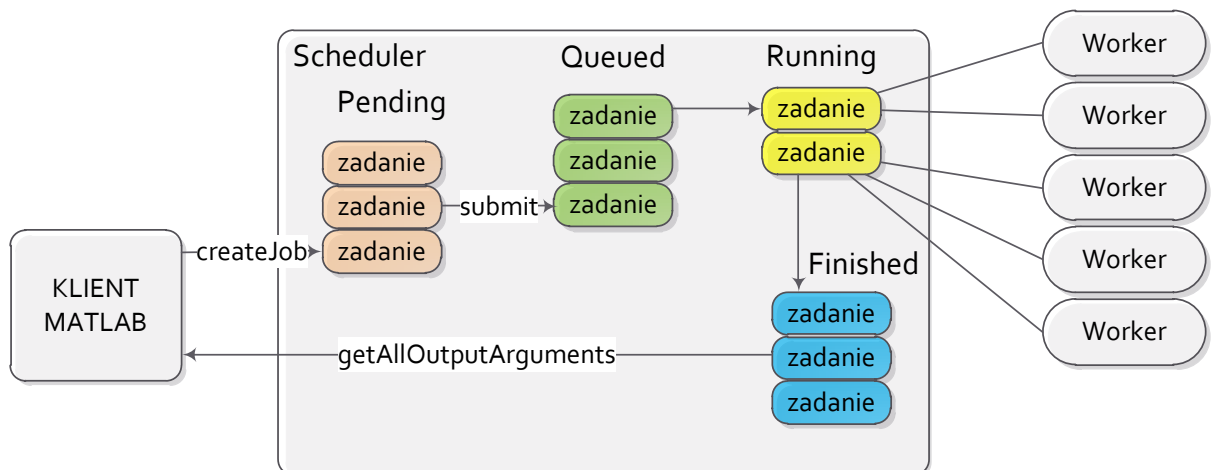
RYSUNEK 59. SCHEMAT REALIZACJI ZADANIA W KLASTRZE MATLAB

Schemat realizacji programu wykonującego obliczenia w środowisku klastrowym MATLAB może zostać opisany następująco. W pierwszej kolejności należy wskazać menedżer zadań, który posłuży do przydzielania zasobów dla tworzonych zadań. Krok ten jest realizowany poprzez wywołanie funkcji `FindResource` z odpowiednimi parametrami. Następnie przy użyciu polecenia `createJob` tworzone jest nowe zadanie, które uzyska status oczekującego (ang. `pending`) na wybranym menedżerze zadań. Przy wykorzystaniu polecenia `createTask` o składni przedstawionej na rysunku 60



RYSUNEK 60. SKŁADNIA POLECENIA `createTask`

tworzone są podzadania wchodzące w skład zadania [133]. Zgłoszenie gotowości realizacji zadania wykonywane jest poprzez wywołanie funkcji `submit`. Zadanie umieszczone zostaje w kolejce zadań gotowych do realizacji (status `queued`). Menedżer zadań szereguje zadania w kolejności ich zgłoszenia (w kolejności wywołania funkcji `submit`). W przypadku, gdy zadanie znajdzie się na szczycie listy zadań, osiąga status uruchomionego (ang. `running`), natomiast scheduler przydziela podzadania składowe do uruchomienia na jednostkach wykonawczych. Jeżeli liczba podzadań w uruchomionym zadaniu jest mniejsza od liczby dostępnych jednostek wykonawczych, scheduler uruchamia kolejne zadanie z kolejki zadań. Po ukończeniu realizacji wszystkich podzadań składowych, zadanie osiąga status zakończony (ang. `finished`). Za pomocą funkcji `getAllOutputArguments` pobierane są wartości zwracane przez wszystkie podzadania składowe. W celu zwolnienia zasobów zajmowanych przez zakończone zadanie wywoływana jest funkcja `destroy`. Na rysunku 61 został przedstawiony cykl życia zadań dla klastrowego środowiska MATLAB [135].



RYSUNEK 61. CYKL ŻYCIA ZADAŃ DLA KLASTROWEGO ŚRODOWISKA MATLAB

5.4. Wykorzystywane biblioteki obliczeniowe

Spośród znacznej liczby bibliotek numerycznych, dostępnych do wykorzystania w celu realizacji obliczeń naukowych, w treści pracy przedstawiono kluczowe biblioteki ze względu na zastosowanie ich w przeprowadzonych obliczeniach dla środowiska MATLAB oraz dla karty graficznej. Podstawową biblioteką numeryczną, z pomocą której wykonywane są fundamentalne operacje na wektorach i macierzach jest biblioteka BLAS (ang. The Basic Linear Algebra Subprograms). Operacje na danych realizowane przez bibliotekę BLAS zostały podzielone na trzy poziomy. Pierwszy poziom stanowią operacje wektorowe, które mogą zostać zapisane w ogólnej postaci [136]:

$$y \leftarrow \alpha x + y \quad (120)$$

Do poziomu pierwszego należy również zaliczyć takie operacje jak wyznaczanie iloczynu skalarnego wektorów, wyszukiwanie minimalnego lub maksymalnego elementu wektora, wyznaczanie sumy wartości bezwzględnych elementów wektora czy wyznaczanie normy wektorowej. Poziom drugi funkcji BLAS realizuje operacje wektorowo-macierzowe. Operacje te mogą zostać przedstawione w ogólnej postaci [136]:

$$y \leftarrow \alpha Ax + \beta y \quad (121)$$

Do poziomu drugiego należy również zaliczyć rozwiązanie układu równań liniowych $Ax = b$ gdzie macierz A posiada postać trójkątną górną lub dolną. Trzeci poziom funkcji BLAS stanowią operacje macierzowe, które mogą zostać wyrażone za pomocą postaci ogólnej [136]:

$$C \leftarrow \alpha AB + \beta C \quad (122)$$

Do poziomu trzeciego należy również zaliczyć rozwiązanie układu równań:

$$AX = \alpha B \text{ lub } XA = \alpha B \quad (123)$$

gdzie macierz A posiada formę trójkątną górną lub dolną. Biblioteka BLAS dla każdego poziomu dostarcza funkcje realizujące obliczenia na danych pojedynczej lub podwójnej precyzji w postaci rzeczywistej lub zespolonej. W celu uzyskania maksymalnej wydajności, biblioteka jest dostarczana w postaci zoptymalizowanej do konkretnej architektury sprzętowej. Kluczowi producenci sprzętu komputerowego wydają własne wersje biblioteki, które zazwyczaj wchodzi w skład udostępnianych bibliotek obliczeniowych (np. Intel Math Kernel Library [137], AMD Core Math Library [138], CUBLAS [139]).

Biblioteką umożliwiającą realizację podstawowych operacji algebry liniowej jest biblioteka LAPACK (ang. Linear Algebra PACKage). W skład operacji możliwych do wykonania z wykorzystaniem biblioteki wchodzi między innymi: rozwiązywanie układu równań liniowych oraz wyznaczenie szeregu dekompozycji macierzy takich jak SVD, EVD, LU, dekompozycja Cholesky'ego, QR. Biblioteka LAPACK wykorzystuje bibliotekę BLAS w celu przeprowadzenia podstawowych operacji na macierzach lub wektorach. Analogicznie do biblioteki BLAS, możliwe jest wykonywanie operacji dla danych pojedynczej i podwójnej precyzji w postaci rzeczywistej lub zespolonej [140]. Dla środowiska karty graficznej oraz środowisk hybrydowych (obliczenia na CPU i GPU) dostępne są biblioteki CULA [76] [141] [142] oraz MAGMA [143] implementujące podstawową funkcjonalność biblioteki LAPACK. W **oddatku A** zebrano wybrane funkcje biblioteki LAPACK oraz ich odpowiedniki dla biblioteki CULA, wykorzystywane przy tworzeniu programów dla potrzeb niniejszej pracy.

Pakiet BOOST jest zbiorem wielu darmowych bibliotek napisanych w języku C++. Poszczególne biblioteki zawierają szereg algorytmów i struktur danych umożliwiających rozwiązywanie konkretnych problemów programistycznych z wykorzystaniem gotowych wzorców. Z bogatego zbioru bibliotek pakietu BOOST (w wersji BOOST 1.32.0 zawartych jest 58 bibliotek) w pracy wykorzystano bibliotekę Boost.Graph implementującą podstawowe algorytmy teorii grafów [144] (procedura `Connected_Components`).

Przybornik eAMOR, opracowany przez M. Rydla, służy do realizacji redukcji modeli metodami FW (ang. Frequency Weighted), FD (ang. Frequency Domain) oraz FMR (ang.

Fourier Model Reduction) [14]. Przybownik wykorzystuje algorytmy ewolucyjne w celu doboru parametrów redukcji. Dla metody FW elementami podlegającymi modyfikacji są rząd oraz częstotliwość graniczna filtrów wagowych dołączanych na wejścia lub wyjścia modelu zredukowanego (w przypadku zastosowania filtrów Butterwortha lub Bessela). Modyfikacja wartości obu parametrów wpływa na zmianę zakresu adekwatności modelu zredukowanego (maksymalnej częstotliwości, do której model poprawnie aproksymuje charakterystyki częstotliwościowe modelu oryginalnego). Do oceny wartości funkcji celu (i pośrednio wartości funkcji przystosowania dla algorytmu ewolucyjnego), przybownik eAMOR wprowadza szereg miar błędu liczonych dla określonej liczby punktów, w dziedzinie częstotliwości, znajdujących się wewnątrz pożądanego przedziału adekwatności [15].

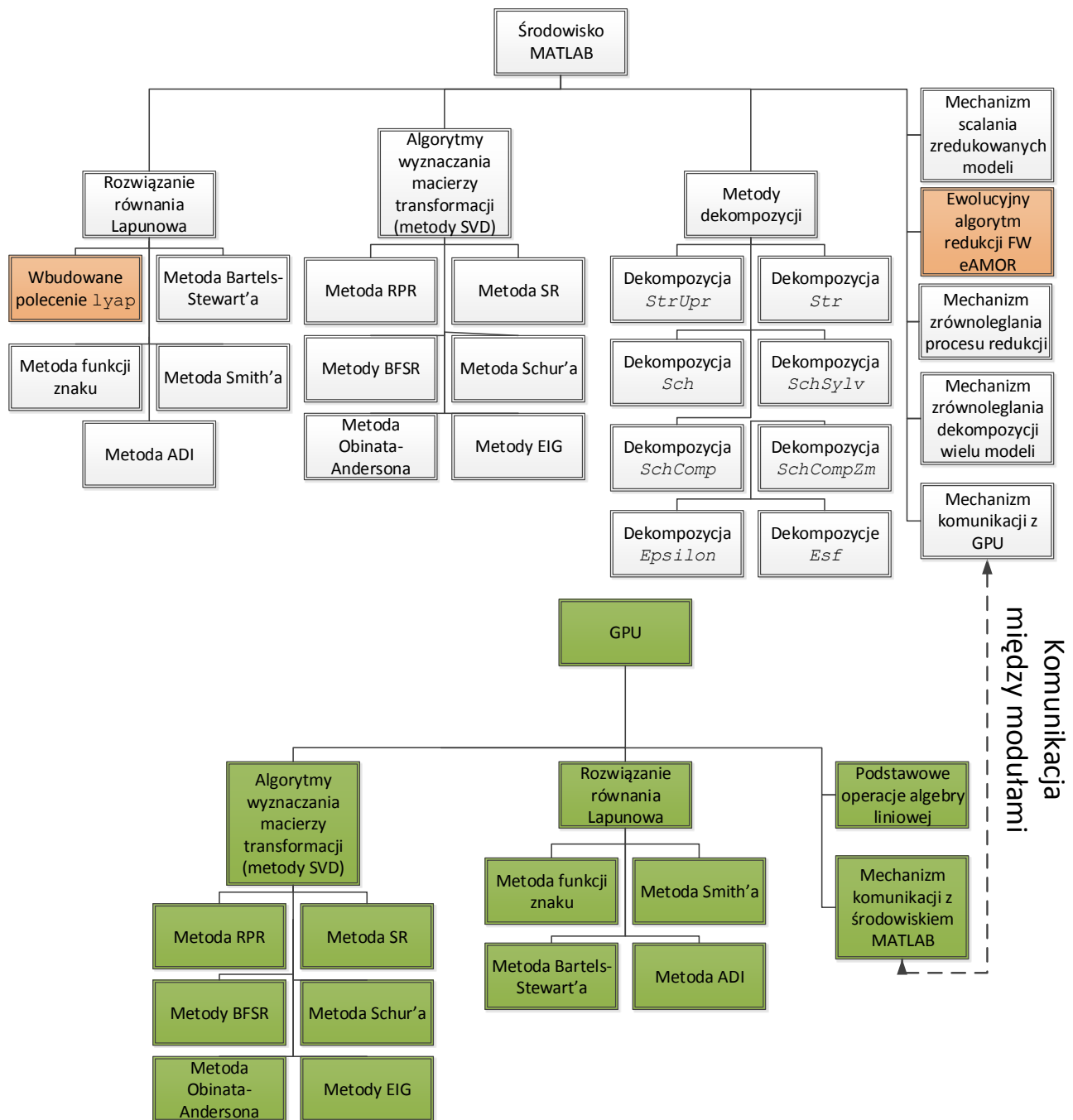
5.5. Zaimplementowana aplikacja obliczeniowa

Struktura zaimplementowanej aplikacji obliczeniowej pod względem funkcjonalności stanowi zbiór modułów przedstawionych na rysunku 62.

Główną częścią zaimplementowanej aplikacji obliczeniowej są procedury i funkcje stworzone w języku MATLAB. Zbiór wybranych funkcji i procedur wraz z ich krótkim opisem został zamieszczony w **dodatku B i C**. Stanowią one rdzeń aplikacji, odpowiedzialny za realizację dekompozycji, wykonywanie obliczeń równoległych na klastrze MATLAB, wyznaczanie macierzy gramianów sterowalności i obserwowalności, wyznaczanie macierzy transformacji z wykorzystaniem metod redukcji SVD oraz komunikację z modułem obliczeń na karcie graficznej. Aplikacja wykorzystuje ewolucyjny algorytm redukcji FW autorstwa dr inż. Marka Rydla z modyfikacjami umożliwiającymi równoległą redukcję modeli. W celu realizacji procesu redukcji z wykorzystaniem GPU na węzłach klastra wyposażonych w kartę graficzną, zaimplementowano moduł komunikacji oparty na wymianie plików z macierzami stanu zredukowanego modelu.

Ogólny schemat wykonania aplikacji można przedstawić następująco. Użytkownik wywołuje aplikację wpisując jej nazwę w konsoli MATLAB oraz podaje szereg parametrów:

- ścieżki do folderów zawierających modele oryginalne,
- ścieżki do folderów, w których zostaną zachowane modele wynikowe,
- typy dekompozycji realizowane na poszczególnych poziomach dla danego modelu,
- liczbę podsystemów, na które zostanie podzielony model na danym poziomie dekompozycji,
- dolny zakres aproksymacji charakterystyk częstotliwościowych modelu zredukowanego (w przypadku dekompozycji slow-fast dolny zakres aproksymacji charakterystyk częstotliwościowych dla części wolnej modelu),
- górny zakres aproksymacji charakterystyk częstotliwościowych modelu zredukowanego (w przypadku dekompozycji slow-fast górny zakres aproksymacji charakterystyk częstotliwościowych dla części szybkiej modelu),
- rozmiar modelu wynikowego (w przypadku dekompozycji slow-fast rozmiar zredukowanej części wolnej modelu),
- dolny zakres aproksymacji charakterystyk częstotliwościowych części szybkiej modelu zredukowanego,
- górny zakres aproksymacji charakterystyk częstotliwościowych części szybkiej modelu zredukowanego,
- rozmiar części szybkiej modelu zredukowanego.

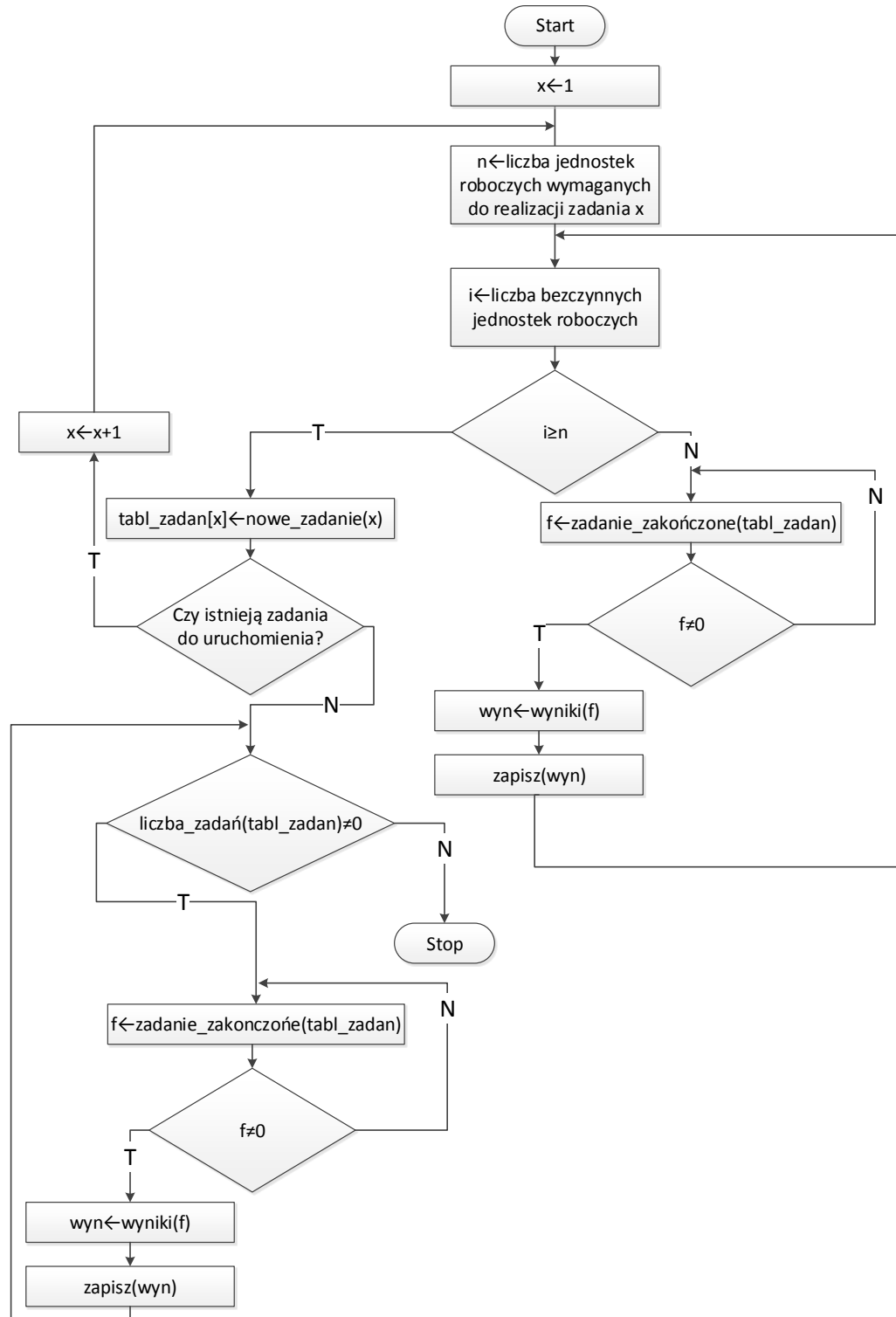


RYSUNEK 62. ZBIÓR MODUŁÓW ZAIMPLEMENTOWANEJ APLIKACJI (kolor szary – moduły zaimplementowane dla środowiska MATLAB, kolor zielony – moduły zaimplementowane dla środowiska procesora graficznego, kolor pomarańczowy – wykorzystywane zewnętrzne programy)

Dla redukcji wielu modeli, parametry mają postać macierzy lub wektorów. Każdy wiersz w przekazanej macierzy/wektorze charakteryzuje proces redukcji z dekompozycją dla konkretnego modelu.

W kolejnym kroku aplikacja dzieli przekazany zbiór danych na pojedyncze zadania (ang. *jobs*) dekompozycji i redukcji, które są przekazywane do realizacji na odrębnych jednostkach roboczych (ang. *workers*). Każde wygenerowane zadanie na tym poziomie dotyczy konkretnego modelu oryginalnego z ustalonym schematem dekompozycji i redukcji. Przydział zadań dla konkretnej jednostki roboczej nie jest narzucony z góry – projektując aplikację założono, że poszczególne węzły klastra mogą mieć różną moc obliczeniową (heterogeniczna architektura równoległa). Wynika to między innymi z ograniczonej liczby

GPU (dwie karty obliczeniowe na cały klaster), różnej liczby rdzeni procesora czy różnej wielkości pamięci RAM. Aplikacja organizuje proces obliczeniowy w taki sposób, aby jak największa liczba jednostek roboczych w danej chwili realizowała obliczenia. W przypadku, gdy dane zadanie zakończy obliczenia, na jego miejsce powoływane jest nowe. Uproszczony schemat modułu zarządzania zadaniami klastra został przedstawiony na rysunku 63.



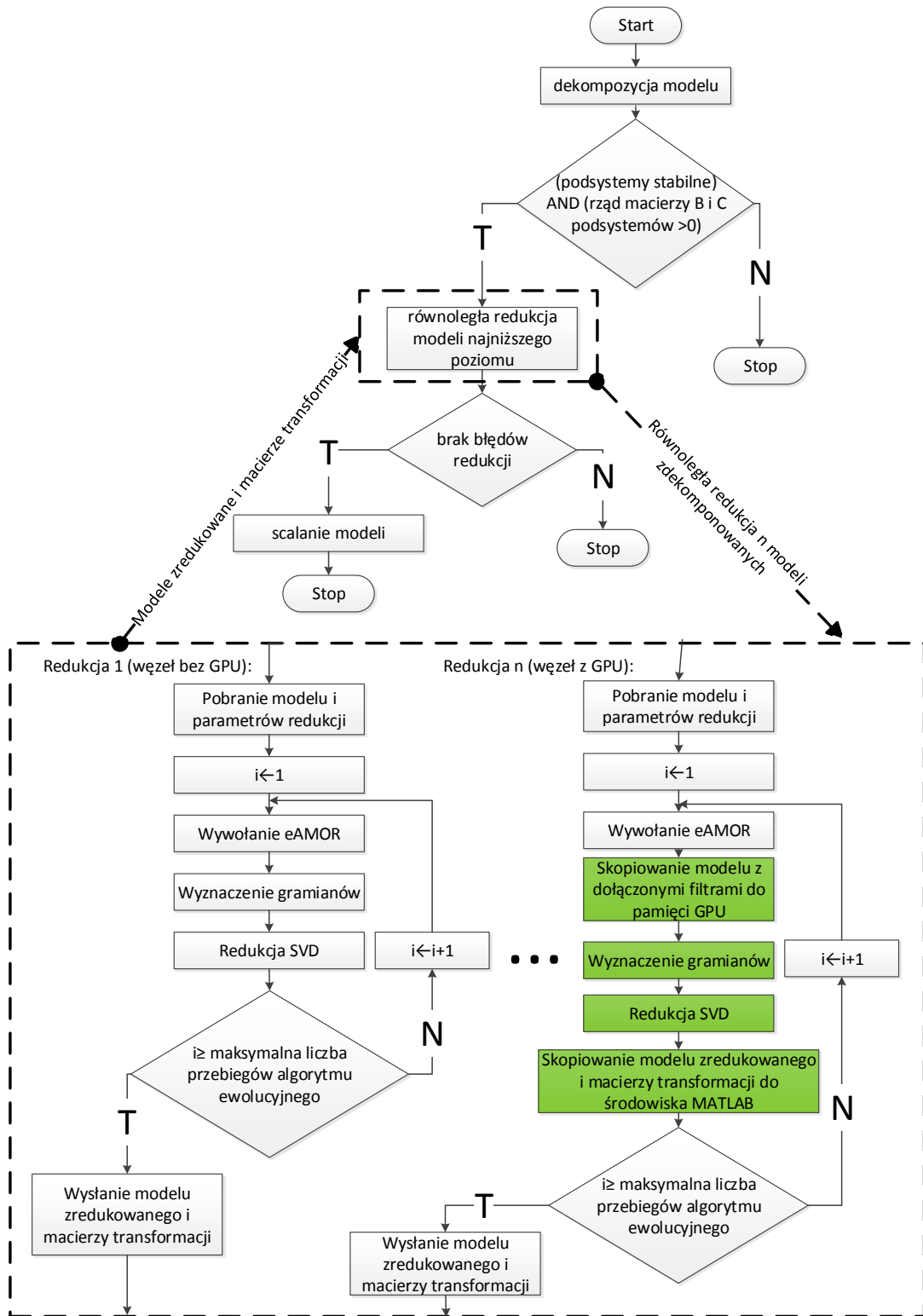
RYСУNEK 63. SCHEMAT DZIAŁANIA MODUŁU ZARZĄDZANIA ZADANIAMI KLASTRA

Realizacja zadania na jednostce roboczej wiąże się z wykonaniem szeregu czynności związanych między innymi z dekompozycją przekazanego modelu na zadaną liczbę podsystemów, równoległą redukcją modeli najniższego poziomu dekompozycji oraz scalaniem otrzymanych modeli zredukowanych z zastosowaniem interfejsów w celu otrzymania modelu wynikowego.

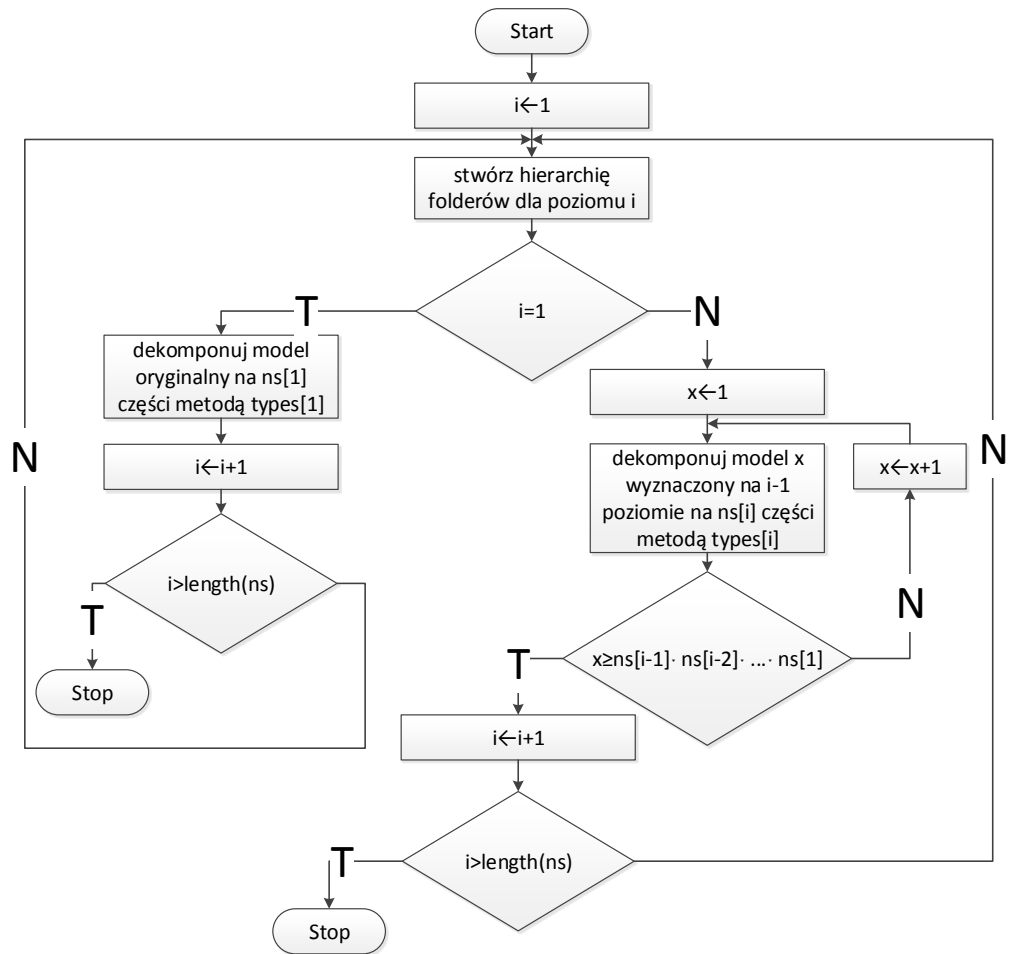
Proces redukcji modeli najniższego poziomu realizowany jest poprzez stworzenie nowego zadania (ang. job) z przyporządkowanymi wątkami (ang. tasks), z których każdy wątek jest odpowiedzialny za redukcję pojedynczego modelu wyznaczonego na najniższym poziomie dekompozycji. Zadanie dodawane jest do menedżera klastra MATLAB, w związku z czym poszczególne procesy redukcji mogą zostać zrealizowane na odrębnych węzłach klastra, lub na jednym węźle przez odrębne jednostki robocze (zgodnie z algorytmem pracy menedżera zadań środowiska MATLAB). W celu realizacji redukcji FW wywoływany jest ewolucyjny algorytm redukcji eAMOR, który wyznacza model z dołączonymi filtrami wyjściowymi. W przypadku, gdy dany węzeł dysponuje jednostką GPU, wynik uzyskany z algorytmu eAMOR poddawany jest redukcji SVD w środowisku karty graficznej. W przypadku, gdy węzeł nie posiada GPU, uzyskany model poddawany jest redukcji z wykorzystaniem zaimplementowanych metod SVD w środowisku MATLAB. Uproszczony proces redukcji pojedynczego modelu został przedstawiony na rysunku 64.

Proces dekompozycji modelu oryginalnego i scalania modeli zredukowanych z zastosowaniem interfejsów realizowany jest zgodnie ze schematami przedstawionymi w rozdziałach 3 i 4. Dla danego przypadku redukcji realizowane są one przez jedną jednostkę roboczą. Dekomponując model na szereg części wydzielanych na określonej liczbie poziomów, aplikacja tworzy hierarchię folderów. Pierwszy poziom w stworzonej hierarchii zawiera pliki stworzone w wyniku procesu dekompozycji dla pierwszego poziomu dekompozycji. Foldery zagnieżdżone zawierają pliki kolejnych poziomów dekompozycji. Ostatni poziom hierarchii folderów zawiera pliki modeli poddawanych redukcji oraz interfejsy między nimi. Proces scalania modeli z interfejsami realizowany jest począwszy od najniższego poziomu hierarchii folderów. W pierwszej kolejności redukowane są modele najniższego poziomu oraz przekształcane są interfejsy między nimi. Aplikacja następnie stopniowo przekształca interfejsy wydzielone na poziomach wyższych, aż do uzyskania modelu wynikowego. Uproszczony schemat blokowy procesu dekompozycji został przedstawiony na rysunku 65. Uproszczony schemat blokowy procesu scalania modeli został przedstawiony na rysunku 66.

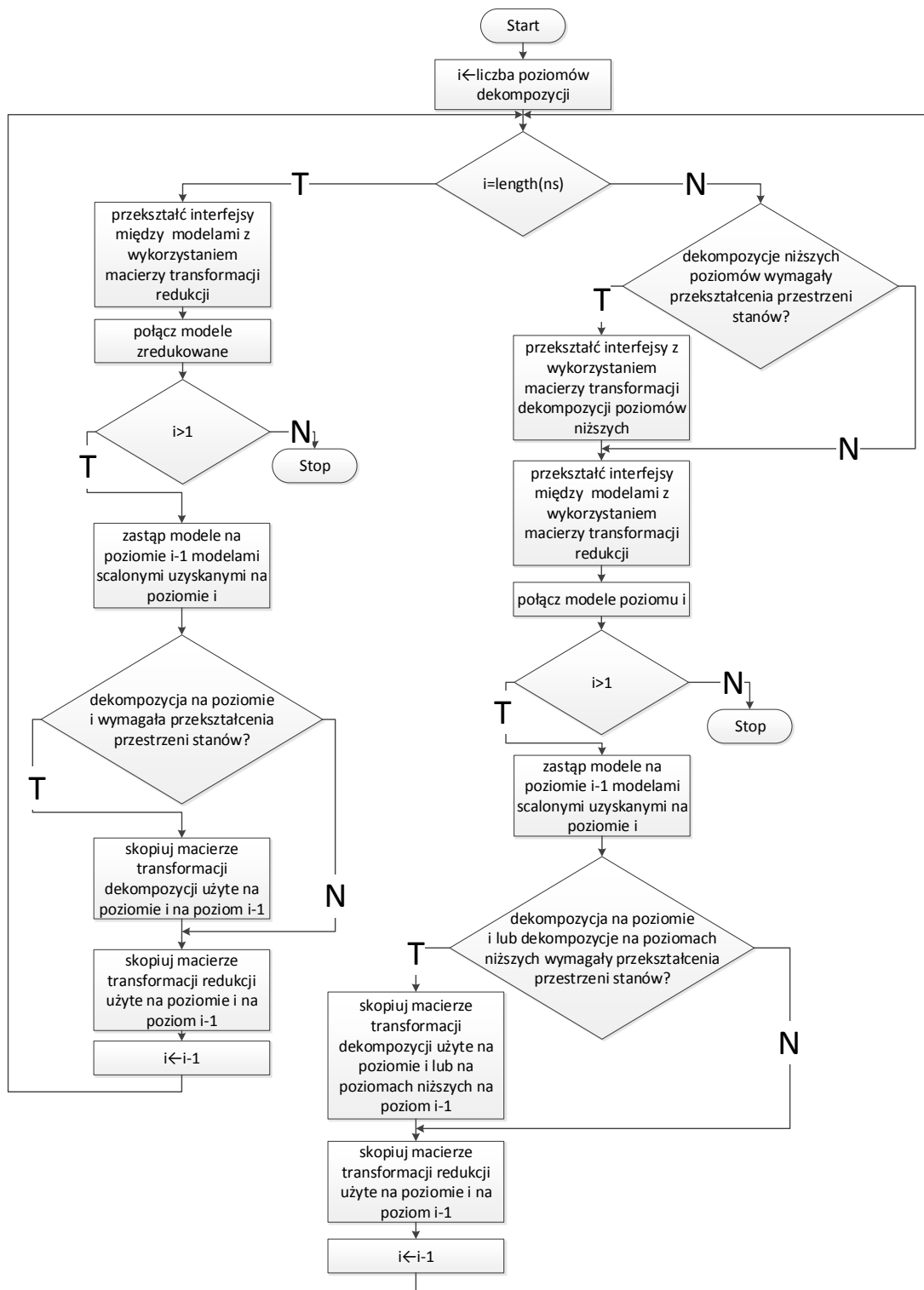
Głównym zadaniem aplikacji zaimplementowanej w języku C++ dla realizacji obliczeń na GPU jest redukcja modeli z wykorzystaniem metod SVD. Wybrane procedury i funkcje wykonujące obliczenia na GPU zostały opisane w **dodatku D, E i F**. Elementy wchodzące w skład modułu obliczeń na GPU realizują między innymi podstawowe operacje algebry macierzowej, rozwiązanie równania Lapunowa, wyznaczanie macierzy transformacji metodami SVD, przesyłanie wyników redukcji do aplikacji uruchomionej w środowisku MATLAB.



RYSUNEK 64. UPROSZCZONY SCHEMAT BLOKOWY PROCESU REDUKCJI MODELI W ŚRODOWISKU KLASTROWYM



RYSUNEK 65. UPROSZCZONY SCHEMAT BLOKOWY PROCESU DEKOMPOZYCJI



RYСУNEK 66. UPROSZCZONY SCHEMAT BLOKOWY SCALANIA ZREDUKOWANYCH MODELII

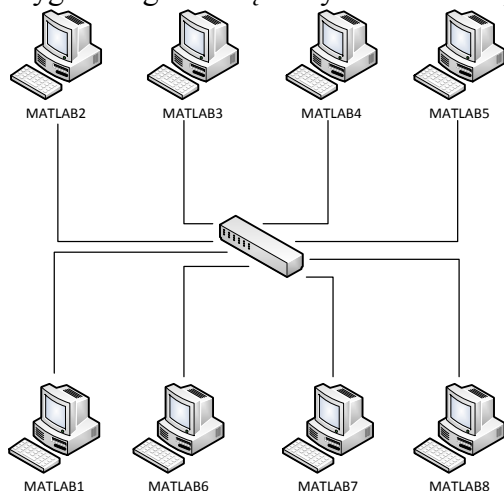
5.6. Charakterystyka środowiska obliczeniowego

Przeprowadzone badania zostały zrealizowane na klastrze ośmiu komputerów PC z zainstalowanym środowiskiem MATLAB. Węzły zostały połączone z wykorzystaniem sieci Ethernet o przepustowości 1Gb/s (standard 1000BASE-T). Struktura klastra została przedstawiona na rysunku 67. W celu oszacowania mocy obliczeniowej poszczególnych jednostek, na każdym węzle zostało uruchomionych dziesięć zadań redukcji modelu

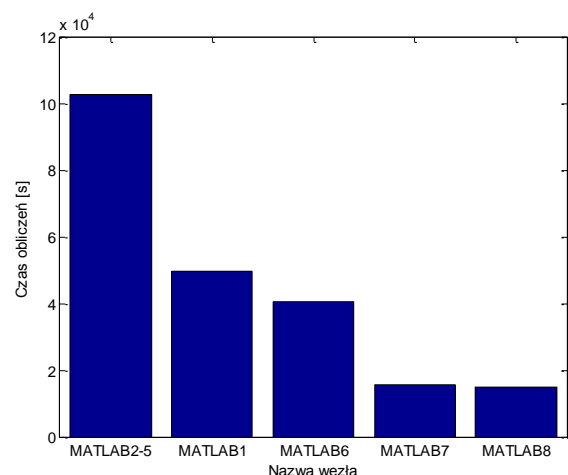
o rozmiarze 6600 zmiennych stanu (model strefy dogrzewu), dla różnych parametrów redukcji. Jako metodę dekompozycji wybrano dekompozycję strukturalną, dwupoziomową (cztery modele najniższego poziomu). Dla algorytmu eAMOR ustawiono limit redukcji dla algorytmu ewolucyjnego na wartość 25 i dla algorytmu przeszukiwania lokalnego na 5. Całkowita liczba redukcji realizowana na modelach o rozmiarze 1650 zmiennych stanu wynosiła zatem $10 \cdot 4 \cdot 30 = 1200$. Na rysunku 68 został przedstawiony wykres czasu potrzebnego na wykonanie obliczeń dla poszczególnych węzłów, gdyby pracowały samodzielnie. W kolejnym kroku uruchomiono te same obliczenia dla różnych konfiguracji klastra. Biorąc pod uwagę różną moc obliczeniową poszczególnych węzłów, postanowiono uzależnić liczbę jednostek roboczych środowiska MATLAB uruchomionych na danym komputerze od liczby rdzeni wchodzących w skład procesora (na każdy rdzeń procesora zostają uruchomione dwie jednostki robocze). Założenie to pozwala w większym stopniu obciążyć komputery dysponujące większą mocą obliczeniową. Wykres przedstawiający czas potrzebny dla realizacji zadania w zależności od konfiguracji klastra został zamieszczony na rysunku 69. Poszczególne konfiguracje klastra zostały zdefiniowane w tabeli 1.

W celu porównania mocy obliczeniowej procesorów graficznych wykonano redukcję modelu rzędu 3300 zmiennych stanu dla modelu bez dekompozycji, z zastosowaniem dekompozycji strukturalnej jednopoziomowej (podział modelu na dwie części) oraz dekompozycji strukturalnej dwupoziomowej (podział modelu na cztery części). Redukcja została wykonana z następującymi parametrami algorytmu eAMOR: łączna liczba redukcji dla algorytmów ewolucyjnych – 700, liczba redukcji dla algorytmu przeszukiwania lokalnego – 100. Czasy redukcji zostały porównane z najwydajniejszą posiadaną jednostką CPU – Intel Core i7 3770, na której uruchomiono analogiczne obliczenia. Na wykresie 70 zamieszczono rezultaty przeprowadzonych obliczeń.

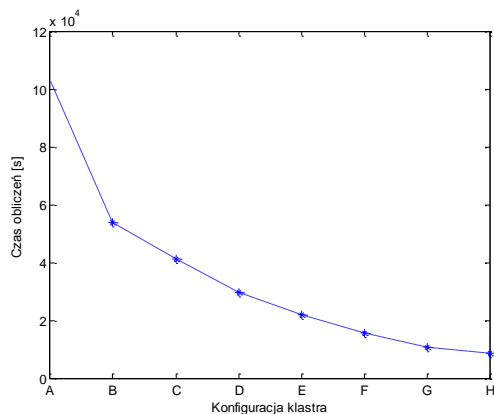
Badania potwierdzają, że wykorzystanie procesorów graficznych daje dobre rezultaty w przypadku masywnych obliczeń na macierzach wysokiego rzędu. Dla modeli niższego rzędu, pomimo skrócenia czasu obliczeń wywołanego redukcją złożoności obliczeniowej dzięki dekompozycji, procesory graficzne radzą sobie gorzej od klasycznych procesorów. W przypadku redukcji modelu bez dekompozycji wyznaczono również opóźnienia nie związane z realizacją procesu obliczeniowego. Wynosiły one łącznie: 5621s na komunikację przybownika eAMOR z aplikacją GPU (wysyłanie modelu oryginalnego z dołączonymi filtrami), 643s na komunikację aplikacji GPU z przybownikiem eAMOR (wysyłanie wyznaczonych macierzy transformacji) oraz 2544s na kopiowanie macierzy modelu oryginalnego z dołączonymi filtrami z pamięci głównej do pamięci GPU.



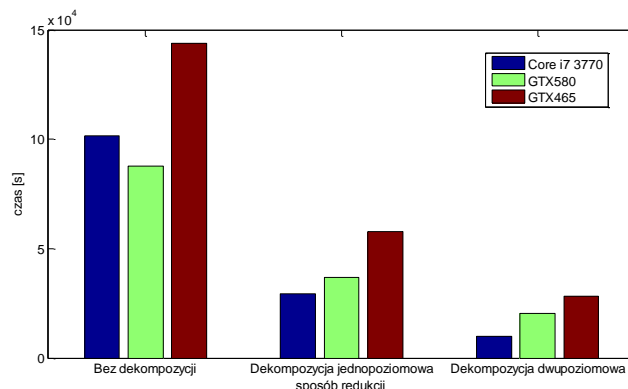
RYSUNEK 67. STRUKTURA KLASTRA OBLICZENIOWEGO



RYSUNEK 68. CZAS REALIZACJI DZIESIĘCIU ZADAŃ REDUKCJI DLA POSZCZEGÓLNYCH KOMPUTERÓW WYKORZYSTYWANYCH W KLASTRZE



RYSUNEK 69. CZAS POTRZEBNY NA REALIZACJĘ DZIESIĘCIU ZADAŃ REDUKCJI W ZALEŻNOŚCI OD KONFIGURACJI KLASTRA



RYSUNEK 70. PORÓWNIANIE CZASU POTRZEBNEGO NA WYKONANIE ZADANIA REDUKCJI DLA GPU I CPU

TABELA 1. KONFIGURACJA KLASTRA

Konfigu- racja	Nazwa węzła								
	MATLAB1	MATLAB2	MATLAB3	MATLAB4	MATLAB5	MATLAB6	MATLAB7	MATLAB8	
A	✗	✓	✗	✗	✗	✗	✗	✗	
B	✗	✓	✓	✗	✗	✗	✗	✗	
C	✗	✓	✓	✓	✗	✗	✗	✗	
D	✗	✓	✓	✓	✓	✗	✗	✗	
E	✓	✓	✓	✓	✓	✗	✗	✗	
F	✓	✓	✓	✓	✓	✓	✗	✗	
G	✓	✓	✓	✓	✓	✓	✓	✗	
H	✓	✓	✓	✓	✓	✓	✓	✓	

5.7. Uwagi

Operacja redukcji modeli LTI wysokiego rzędu, poza opracowaniem szeregu algorytmów, wymaga także zaprojektowania i realizacji środowiska obliczeniowego, w którym będzie możliwa równoległa realizacja opracowanych algorytmów. Operacja redukcji rzędu modeli z zastosowaniem SVD charakteryzuje się wysoką złożonością obliczeniową ($O(n^3)$), co pociąga za sobą długie czasy obliczeń. Zastosowanie metody FW wymaga doboru parametrów filtrów częstotliwościowych w celu zapewnienia niskiego błędu aproksymacji charakterystyk częstotliwościowych w zadanym przedziale częstotliwości, co przy zastosowaniu algorytmu ewolucyjnego znacznie wydłuża czas obliczeń. Powoduje to, że redukcja modeli LTI wysokiego rzędu wymaga znacznej mocy obliczeniowej, jaką jeszcze kilkanaście lat temu posiadały superkomputery.

Dla środowiska MATLAB, poczynając od wersji 2008b, zaczęto wprowadzać obliczenia równoległe z wykorzystaniem procesorów wielordzeniowych, a także możliwość obliczeń rozproszonych z wykorzystaniem klastrów komputerowych. W tym samym czasie stały się dostępne procesory graficzne zawierające setki rdzeni obliczeniowych, wykonujących obliczenia zmiennoprzecinkowe.

W ramach realizowanej pracy doktorskiej zaprojektowano oraz zrealizowano heterogeniczny system obliczeń równoległych, zawierający szereg komputerów PC z procesorami wielordzeniowymi oraz procesorami graficznymi, który może być następnie

praktycznie dowolnie rozbudowany. Centralnym środowiskiem, pozwalającym na podział zadań w klastrze komputerowym, jest MATLAB.

Głównym zadaniem autora było opracowanie algorytmów oraz programów w środowisku MATLAB oraz procesorów graficznych, w celu równoległej realizacji operacji redukcji. W szerokim zakresie wykorzystane były programy opracowane w wielu dostępnych bibliotekach i pakietach: BLAS, LAPACK, CUBLAS, CUDA, CULA, BOOST, eAMOR. Szereg programów autor opracował samodzielnie ze względu na brak odpowiednich programów tak w środowisku MATLAB, jak i dla procesora graficznego. W środowisku MATLAB zaimplementowane zostały programy realizujące szereg dekompozycji modeli LTI: *Str*, *StrUpr*, *Sch*, *SchSylv*, *SchComp*, *SchCompZm*, *Esf-SchComp*, *Esf-SchSylv*, *Esf-StrUpr*, *Esf-Str*, *Esf-Sch*, *Esf-MLS*, *Esf-SchCompZm*. W celu realizacji dekompozycji slow-fast, dla modeli podsystemów parownika kotła energetycznego BP-1150, opracowana została modyfikacja dekompozycji Epsilon. Stworzono programy realizujące wielopoziomą dekompozycję modeli LTI z wykorzystaniem wymienionych metod (dla analizowanych modeli zastosowano dekompozycję dwupoziomą). Zaimplementowano programy umożliwiające wyznaczenie macierzy gramianów sterowalności i obserwowalności metodami: Bartels-Stewart'a, Smith'a, funkcji znaku, ADI dla środowiska MATLAB oraz dla środowiska procesora graficznego. Opracowano programy wyznaczające macierze transformacji redukcji dla metod: RPR, SR, BFSR, Schur'a, Obinata-Andersona, EIG-BFSR dla środowiska MATLAB oraz dla środowiska procesora graficznego. Zmodyfikowano przybornik eAMOR autorstwa dr Marka Rydla dla celów redukcji metodą FW z wykorzystaniem środowiska klastrowego oraz procesorów graficznych. Połączono następnie wszystkie zaimplementowane programy tworząc system obliczeniowy, który realizuje redukcję równoległą złożonych modeli LTI. W opracowanym systemie obliczeniowym jest możliwa realizacja redukcji modeli praktycznie dowolnego rzędu w akceptowalnym czasie.

6. Modele matematyczne podsystemów kotła energetycznego BP-1150 i ich dekompozycja

W rozdziale przedstawiono modele matematyczne podsystemów parownika kotła energetycznego BP-1150 na podstawie pracy [32].

6.1. Kocioł energetyczny BP-1150

Kotły BB-1150 oraz BP-1150 wchodzące w skład bloków energetycznych o mocy 360 MW są kotłami przepływowymi opartymi na konstrukcji firmy SULZER, ze stałym punktem końcowym strefy odparowania. Separacja wody od pary na wylocie z parownika odbywa się w pionowym wodooddzielaczu. Kotły przystosowane są do pracy w układzie blokowym z turbiną parową o mocy znamionowej 360 MW. W zakresie obciążeń 30 – 85 % blok pracuje przy ciśnieniu poślizgowym, natomiast dla obciążeń większych od 85 % praca bloku odbywa się przy stałym ciśnieniu. Kotły BB-1150 oraz BP-1150 są kotłami jednociągowymi o wysokości ok. 100 m. Aktualnie w Polskim Systemie Energetycznym pracuje ok. dwudziestu bloków wyposażonych w kotły BB-1150 oraz BP-1150 (Bełchatów, Opole). Usytuowanie poszczególnych urządzeń w kotle, ze szczególnym uwzględnieniem wymienników ciepła przedstawiono na rys. 71 [32].

Kocioł BP-1150 opalany jest węglem kamiennym, który po zmieleniu w młynach misowo-rolkowych wdmuchiwany jest w postaci pyłu do komory paleniskowej przez dysze czterech narożnikowych palników pyłowych. Powietrze do spalania dostarczane jest przez jeden wentylator powietrza pierwotnego oraz dwa wentylatory powietrza wtórnego. Powietrze pierwotne, przeznaczone do suszenia i transportu pyłu węglowego jest podgrzewane do temperatury ok. 350 °C w obrotowym podgrzewaczu powietrza młynowego oraz wdmuchiwane do młynów przez wentylatory młynowe. Powietrze wtórne, podgrzewane w dwóch obrotowych podgrzewaczach powietrza, dostarczane jest wprost do komory paleniskowej kotła [32].

Komora paleniskowa, usytuowany nad nią kanał konwekcyjny oraz strop wykonane są ze ścian membranowych szczelnie spawanych. Przekrój komory paleniskowej wynosi 14.3 × 15.7 m.

Podstawowe dane techniczne kotła BP-1150 są następujące [32]:

- wydajność maksymalna trwała 1150 [ton/godz] ≈ 320 [kg/sek]
- wydajność cieplna 750 [Gcal/godz] ≈ 873 [MW]
- ciśnienie pary na wylocie z przegrzewacza pierwotnego 18.3 [MPa]
- temperatura pary przegrzanej 540 ± 5 [°C]
- temperatura wody zasilającej 255 [°C]
- ciśnienie w separatorze 20.3 [MPa]
- ciśnienie pary wtórnie przegrzanej 4.5/4.3 [MPa]
- temperatura pary wtórnie przegrzanej 335/540 ± 5 [°C]

Oznaczenia na rys. 71 są następujące [32]:

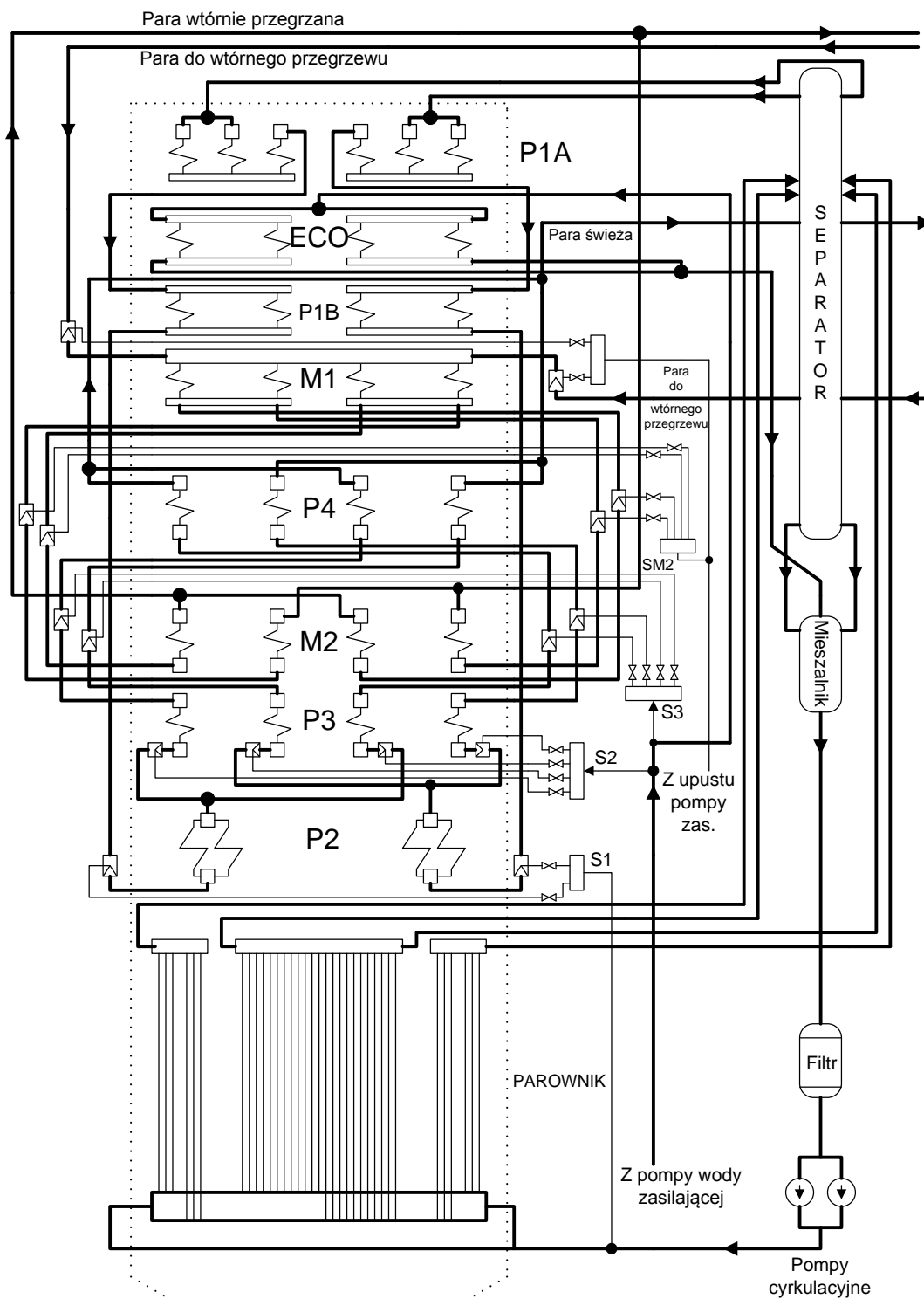
- P1A – wewnętrzne rury wieszakowe,
- P1B – przegrzewacz konwekcyjny wysokiego ciśnienia;
temp. pary - 367/388 °C; temp. spalin - 600/455 °C,
- P2 – przegrzewacz naścienny opromieniowany;
temp. pary - 378/414 °C,
- P3 – przegrzewacz grodziowy;
temp. pary - 414/500 °C; temp. Spalin - 1200/1000 °C,
- P4 – przegrzewacz końcowy pary świeżej;
temp. pary - 480/540 °C; temp. Spalin - 900/780 °C,

- M1 – przegrzewacz początkowy niskiego ciśnienia (wtórny);
temp. pary - 335/463 °C; temp. Spalin - 780/600 °C,
- M2 – przegrzewacz końcowy niskiego ciśnienia (wtórny);
temp. pary - 463/540 °C; temp. Spalin - 1000/900 °C,
- ECO – podgrzewacz wody;
temp. wody - 255/285 °C; temp. spalin - 455/342 °C.

W skład parownika kotła BP-1150 wchodzi: pionowe rury ekranowe parownika stanowiące ekran komory paleniskowej, separator, mieszalnik, filtr, pompa cyrkulacyjna oraz rurociągi łączące (rys. 72). Cały ekran komory paleniskowej jest podzielony na cztery szczelnie spawane ściany. Szczelność ścian uzyskano przez połączenie sąsiednich rur ekranowych wspawaną pletwą. Ekran komory paleniskowej został podzielony na sekcje w celu właściwego doboru przepływu czynnika roboczego. Każda sekcja zasilana jest z kolektora pierścieniowego przez rury łączące. W poszczególnych rurach łączących określono właściwy przepływ czynnika poprzez wspawanie kryz dławiących [32].

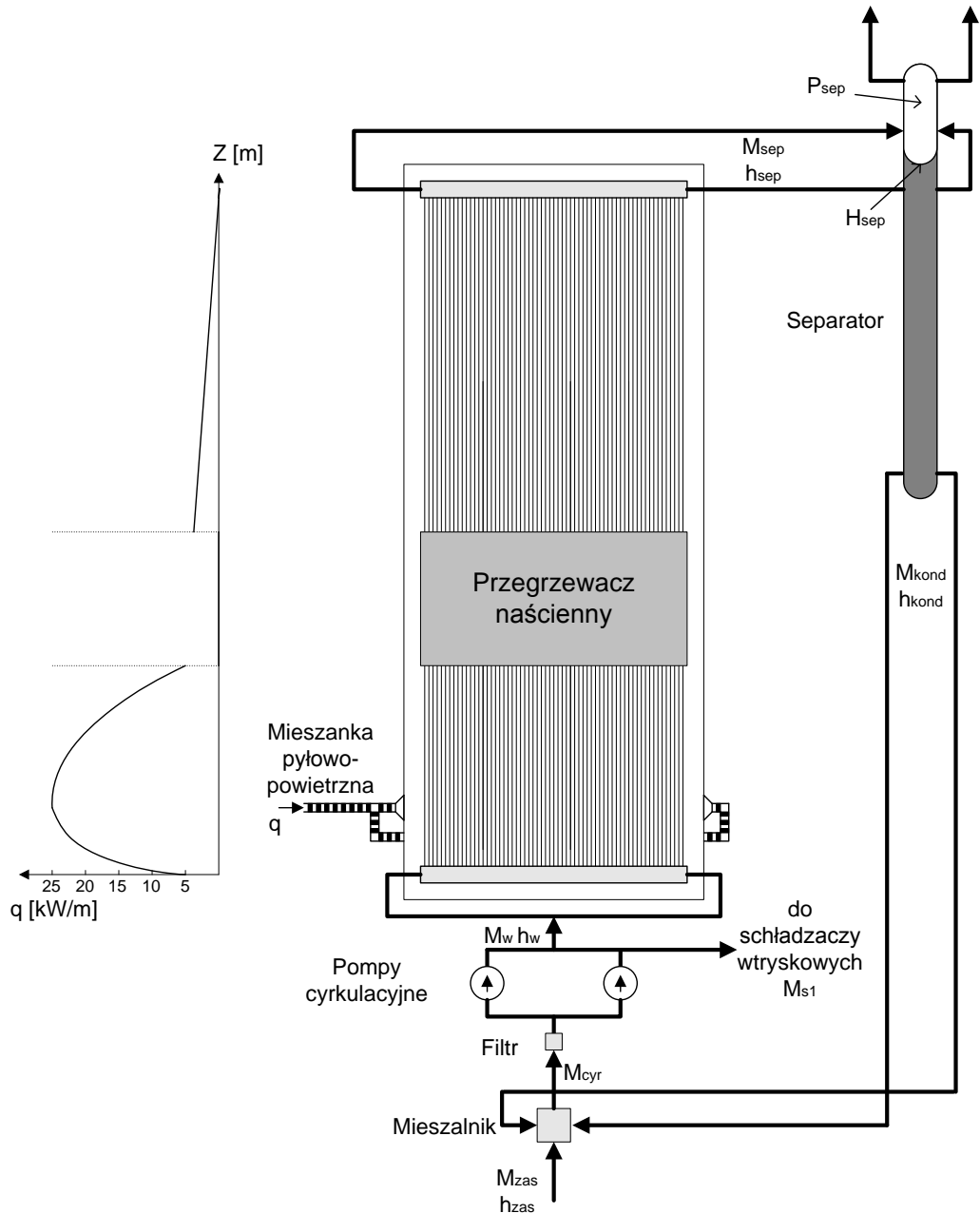
Do poziomu 51.4 m. ekrany parownika wykonane są z rur Φ 30*5 o podziałce 44.5 mm. Na ścianach bocznych liczba rur równoległych wynosi $320 \times 2 = 640$ szt., natomiast na ścianie przedniej i tylnej $352 \times 2 = 704$ szt. W pasie najwyższych obciążeń cieplnych (od poziomu 23.1 m. do 36.2 m.) ekrany wykonane są ze stali 15 HM ze względu na możliwość wystąpienia odparowania filmowego. Na poziomie 51.4 m. na wszystkich ścianach ekranowych wykonane są trójniki. Powyżej linii trójnikowania ściany kotła wykonane są z rur $\Phi 44.5 \times 5$ o podziałce 89 mm. Na rys. 72 przedstawiono przybliżony rozkład strumienia cieplnego wzdłuż rur parownika dla sekcji umieszczonych w środkowej części ścian ekranu. Maksymalny strumień cieplny występuje w pasie palników pyłowych i przy maksymalnym obciążeniu wynosi ok. 25 [kW/m] (w odniesieniu do 1 m. pojedynczej rury). Między poziomem 36.5 m. a 47.3 m. rury ekranowe parownika są przykryte przegrzewaczem naściennym, co powoduje że w pasie tym strumień cieplny dopływający do rur parownika wynosi zero [32].

Para mokra po opuszczeniu górnych kolektorów ekranowych płynie czterema rurociągami Φ 324*36 i wpływa do wodooddzielacza czterema króćcami wspawanymi w jego górnej części. Wodooddzielacz o długości całkowitej 33.8 m. usytuowany jest pionowo między poziomem 57.4 m. a 91 m. Średnica wewnętrzna separatora wynosi 800 mm, a grubość ścianki 50 - 70 mm. Króćce wlotowe usytuowane są stycznie do separatora z lekkim odchyleniem w dół, co powoduje powstanie wiru i separację pary od wody. W górnej części separatora wspawano stycznie dwa króćce wylotowe pary Φ 356, natomiast w dolnej części dwa króćce odprowadzające wodę do mieszalnika Φ 406. Pomiaru poziomu wody w separatorze dokonuje się poprzez pomiar różnicy ciśnień statycznych pomiędzy wyprowadzeniami wykonanymi w dolnej i górnej części separatora. Nominalna wysokość wody w separatorze wynosi 12 m. (mierzona względem dna separatora) [32].



RYSUNEK 71. UPROSZCZONY SCHEMAT TECHNOLOGICZNY KOTŁA BP-1150 W PRZEKROJU

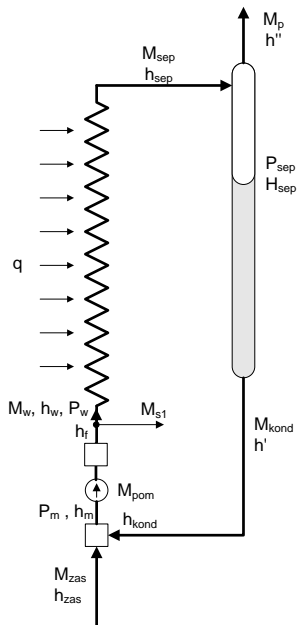
W mieszalniku o wymiarach Φ 930*65 i długości całkowitej 2.8 m. odbywa się mieszanie ze sobą dwóch strumieni wody: wody zasilającej płynącej z podgrzewacza wody oraz kondensatu płynącego z separatora. Z mieszalnika woda przepływa do filtra o wymiarach Φ 930*65 i długości całkowitej 4.6 m. W filtrze umieszczony jest wkład sitowy z otworami Φ 4 [32].



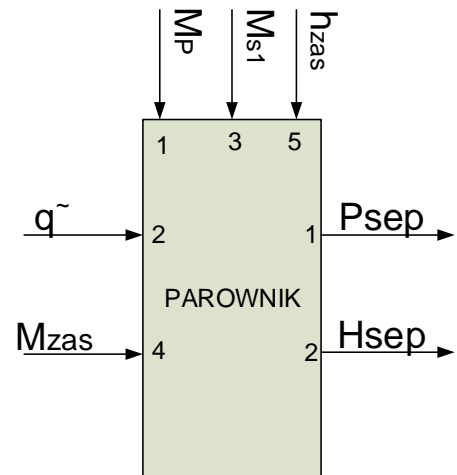
RYSUNEK 72. UPROSZCZONY SCHEMAT PAROWNIKA KOTŁA BP-1150

6.2. Dekompozycja fizyczna modelu matematycznego parownika kotła BP-1150

Uproszczony schemat przepływu wody i pary w parowniku kotła BP-1150 przedstawiono na rys. 73 Schemat ten jest podstawą do opracowania modelu parownika (rys. 74). Model parownika zawiera modele następujących podsystemów: mieszalnik, filtr, ekran komory paleniskowej, separator, pompa cyrkulacyjna, węzeł zasilania schładzaczy pary, rurociągi łączące [32].



RYSUNEK 73. UPROSZCZONY SCHEMAT PRZEPLYWU WODY I PARY W PAROWNIKU KOTŁA BP-1150



RYSUNEK 74. MODEL PAROWNIKA

Poszczególne podsystemy modelu parownika mogą być opisane poprzez podanie wektorów wejściowych U_2^i , wyjściowych Y_2^i oraz określenie operatorów opisujących poszczególne podsystemy F_2^i [32]:

Ekran komory paleniskowej

$$[h_{sep} \ M_{sep} \ P_w]^T = \text{parownik. ekran} \left([M_w \ q \ h_w \ P_{sep}]^T \right)$$

Separator

$$[P_{sep} \ H_{sep}]^T = \text{parownik. separator} \left([M_{kond} \ M_{sep} \ h_{sep} \ M_p]^T \right)$$

Mieszalnik

$$[M_{kond} \ h_m]^T = \text{parownik. mieszalnik} \left([M_{pom} \ M_{zas} \ h_{zas} \ h_{kond}]^T \right)$$

Filtr

$$h_f = \text{parownik. filtr} \left([M_{pom} \ h_{we_f}]^T \right) \quad (124)$$

Pompa cyrkulacyjna

$$M_{pom} = \text{parownik. pompa} \left([P_{sep} \ P_w \ H_{sep} \ M_{kond}]^T \right)$$

Węzeł zasilania schładzaczy pary

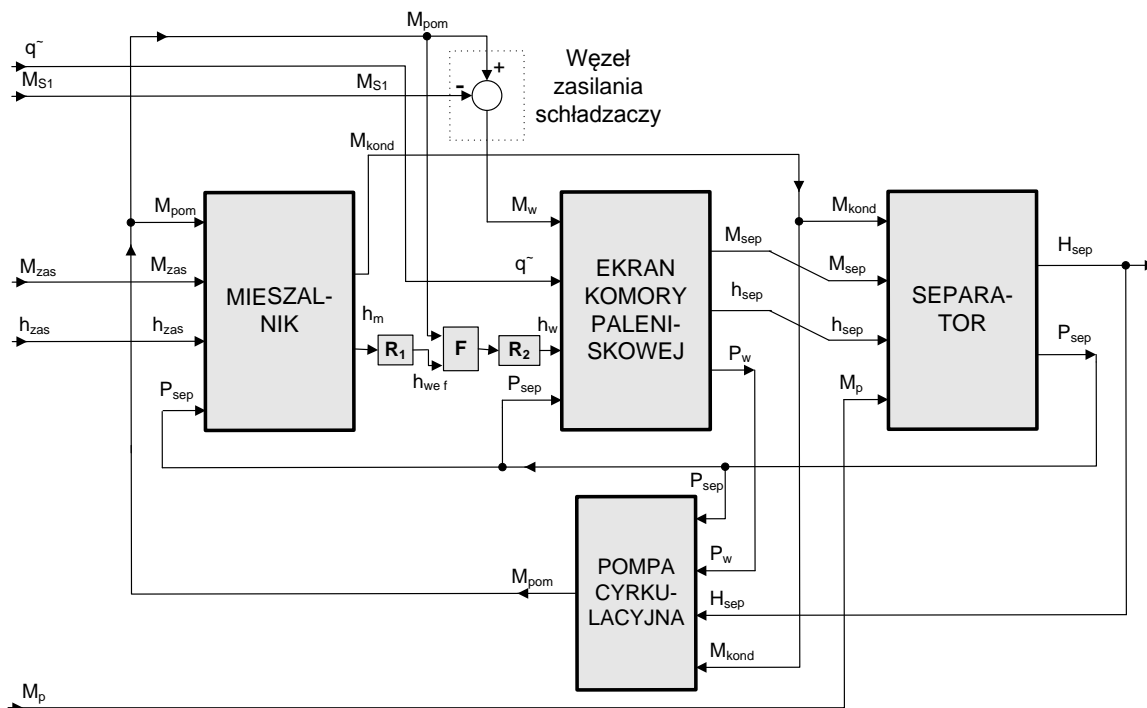
$$M_w = \text{parownik. węzeł_zasilania_schładzaczy} \left([M_{pom} \ M_{s1}]^T \right)$$

Rurociągi łączące

$$h_{we_f} = \text{parownik. rurociąg1} (h_m)$$

$$h_w = \text{parownik. rurociąg2} (h_f)$$

Na rys. 75 przedstawiono model parownika kotła BP-1150 w postaci schematu blokowego.



RYSUNEK 75. STRUKTURA MODELU PAROWNIKA KOTŁA BP-1150 (R1, R2 -RUROCIĄGI ŁĄCZĄCE, F - FILTR)

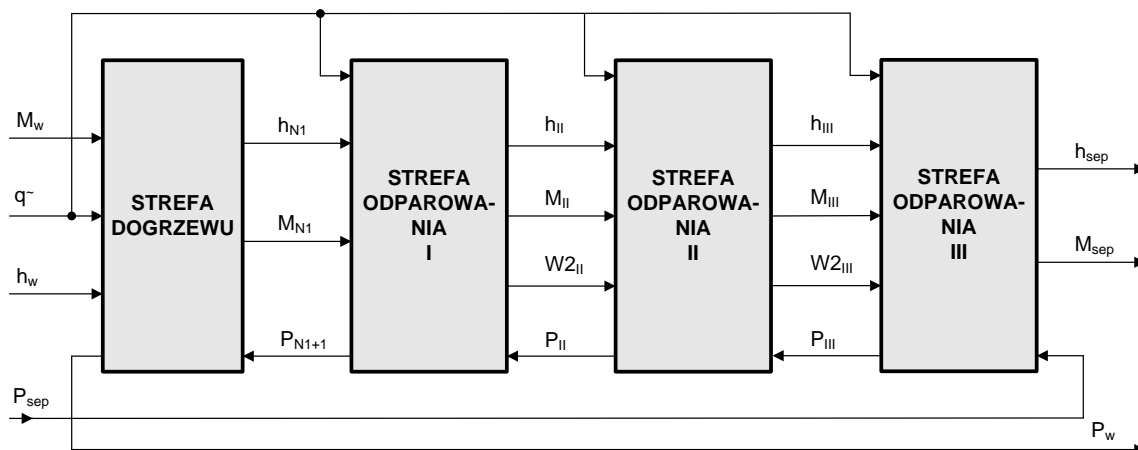
Podsystemy modelu parownika: MIESZALNIK, SEPARATOR, POMPA CYRKULACYJNA, RUROCIĄGI ŁĄCZĄCE nie zawierają podsystemów niższego poziomu i opisane są układami równań różniczkowych, przedstawionymi w dalszej części pracy. Natomiast w modelu ekranu komory paleniskowej wyróżniane są trzy podsystemy: STREFA DOGRZEWU oraz STREFA ODPAROWANIA I, STREFA ODPAROWANIA II i STREFA ODPAROWANIA III. Strukturę modelu rur ekranowych parownika kotła BP-1150 w postaci schematu blokowego przedstawiono na rys 76.

W modelu rur ekranowych parownika można wyróżnić trzy podsystemy o istotnie różniących się właściwościach [32]:

- strefa dogrzewu rozciągająca się od przekroju wlotowego rur ekranowych do przekroju, w którym rozpoczyna się objętościowe wrzenie wody (entalpia wody osiąga entalpię $h'(P)$ - przekrój współrzędnej Z_I ,
- strefa odparowania I z dużym obciążeniem cieplnym w której występuje intensywna generacja pary, rozciągająca się od przekroju o współrzędnej Z_I do miejsca gdzie rury ekranowe parownika są przykryte przez przegrzewacz naścienny (poziom 36.5 m.),
- strefa odparowania II z zerowym obciążeniem cieplnym ze względu na przykrycie przegrzewaczem naściennym,
- strefa odparowania III z niewielkim obciążeniem cieplnym oraz mniejszą prędkością przepływu czynnika (rury ekranowe o większej średnicy).

Przekrój graniczny między strefą dogrzewu i odparowania posiada zmienne położenie Z_I , oraz przemieszcza się z prędkością U_I . Jednak dla modelu zlinearyzowanego, słusznego dla małych odchyleń od stanu ustalonego, przemieszczenia przekroju granicznego są niewielkie i mogą być zaniedbane [32].

Poszczególne wielkości z indeksem *NI* określają parametry przepływu wrzącej wody w przekroju granicznym między strefami. Wielkości z indeksami *II* oraz *III* dotyczą wlotu do strefy odparowania II oraz III, *sep* - wlotu do separatora, natomiast *w* - wlotu do rur ekranowych.



RYSUNEK 76. STRUKTURA MODELU RUR EKRANOWYCH KOMORY PALENISKOWEJ PAROWNIKA KOTŁA BP-1150

Opis poszczególnych podsystemów ekranu komory paleniskowej jest następujący:

Strefa dogrzewu:

$$[h_{N1} \ M_{N1} \ P_w]^T = \text{parownik.ekran.dog} ([M_w \ q \ h_w \ P_{N1+1}]^T)$$

Strefa odparowania I:

$$[M_{II} \ h_{II} \ W2_{II} \ P_{N1+1}]^T = \text{parownik.ekran.odpI} ([q \ h_{N1} \ M_{N1} \ P_{II}]^T)$$

Strefa odparowania II:

$$[M_{III} \ h_{III} \ W2_{III} \ P_{II}]^T = \text{parownik.ekran.odpII} ([q \ M_{II} \ h_{II} \ P_{III} \ W2_{II}]^T)$$

Strefa odparowania III:

$$[M_{sep} \ h_{sep} \ P_{III}]^T = \text{parownik.ekran.odpIII} ([q \ M_{III} \ h_{III} \ P_{sep} \ W2_{III}]^T)$$

(125)

6.3. Modele matematyczne podsystemów parownika

W rozdziale przedstawiono równania bilansowe masy, energii, pędu, a także równia przewodnictwa cieplnego w elementach stalowych.

6.3.1. Rury ekranowe

W rurach ekranowych parownika przepływowego czynnik roboczy przepływa w znacznej liczbie równoległych rur połączonych kolektorami dolnym i górnym. Z punktu widzenia dynamiki kotła przepływowego istotny jest efekt sumaryczny procesów termodynamicznych zachodzących w poszczególnych rurach, ujawniający się w zmianach parametrów czynnika roboczego w kolektorach dolnym i górnym parownika przepływowego. Z tego powodu, przy opracowaniu modelu matematycznego parownika przepływowego wzięto pod uwagę jedną zastępczą rurę parownika o średnim obciążeniu cieplnym i średnim przepływie masowym czynnika roboczego. Do analizy właściwości dynamicznych parownika przepływowego pracującego przy ciśnieniu podkrytycznym przyjęto model strefowy [32].

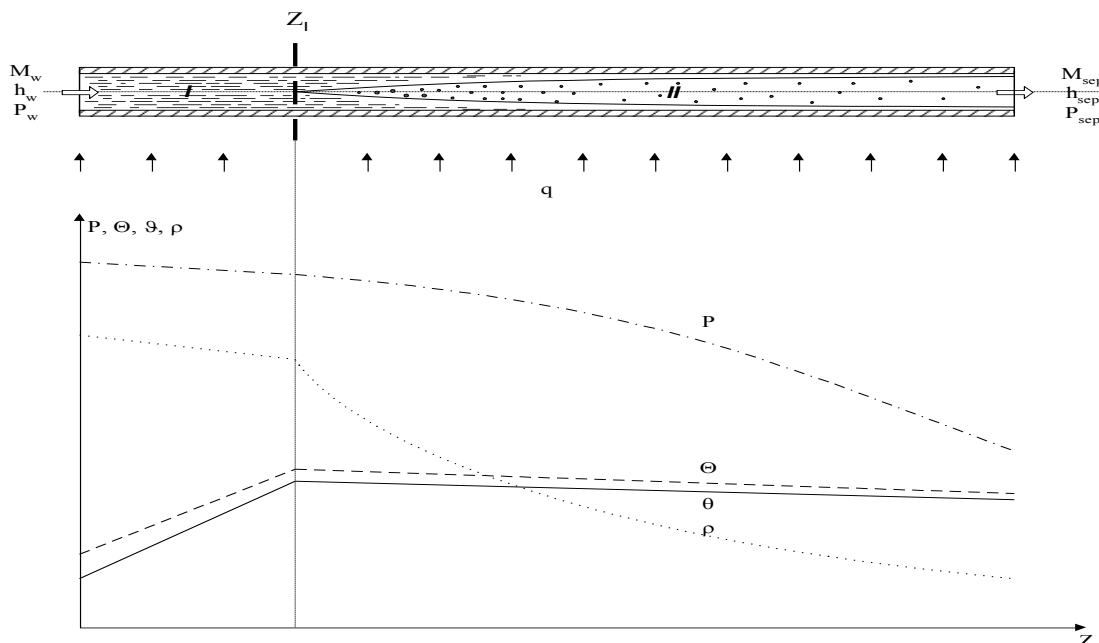
Ze względu na stan czynnika roboczego w modelu parownika kotła BP-1150 wyróżniono dwie strefy [32]:

I - strefę dogrzewu wody do temperatury nasycenia,

II - strefę odparowania wody.

Woda zasilająca parownik ma temperaturę mniejszą od temperatury nasycenia, odpowiadającej ciśnieniu panującemu w parowniku. Z tego powodu na początkowym odcinku rury ekranowej parownika następuje dogrzew wody do temperatury nasycenia. Odcinek ten nosi nazwę strefy dogrzewu. Przyjęto, że przekrojem granicznym strefy dogrzewu jest przekrój, w którym rozpoczyna się proces objętościowego wrzenia wody

(entalpia czynnika roboczego przyjmuje wartość entalpii wrzącej wody, odpowiadającej ciśnieniu panującemu w tym przekroju). Począwszy od tego przekroju (oznaczonego na rys. 77 współrzędną Z_1) następuje proces stopniowego odparowania wody. Bardzo istotną cechą charakterystyczną parownika przepływowego są zmiany położenia przekroju granicznego między strefą dogrzewu i odparowania przy zmieniających się warunkach pracy parownika (zmiany strumienia cieplnego, przepływu wody zasilającej, entalpii wlotowej, itd.) [32].



RYSUNEK 77. MODEL STREFOWY PAROWNIKA PRZEPLYWOWEGO PRZY USTALONYM KOŃCU STREFY ODPAROWANIA ORAZ PODSTAWOWE PARAMETRY CZYNNIKA ROBOCZEGO

Na podstawie pracy [32], równania różniczkowe cząstkowe opisujące jednowymiarowy przepływ wody w strefie dogrzewu, otrzymane na podstawie bilansu masy, energii oraz pędu, przyjmują następującą postać różniczkową:

$$\begin{aligned} \frac{\partial M}{\partial z} &= -\frac{\partial \rho}{\partial t} \\ \rho \frac{\partial h}{\partial t} + M \frac{\partial h}{\partial z} - \frac{\partial P}{\partial t} - \frac{M}{\rho} \frac{\partial P}{\partial z} &= q'_d \quad (126) \\ \frac{\partial M}{\partial t} + 2 \frac{M}{\rho} \frac{\partial M}{\partial z} - \frac{M^2}{\rho^2} \frac{\partial \rho}{\partial z} + \rho g \cos \beta + 10^3 \frac{\partial P}{\partial z} + f \frac{M^2}{\rho} &= 0 \end{aligned}$$

gdzie:

$$M = M(z, t) = \rho w, \quad \rho = \rho(z, t) = \rho(P, h), \quad h = h(z, t), \quad P = P(z, t), \quad q'_d = \frac{4(\Theta_w - \Theta)\alpha}{d},$$

$$\Theta = \Theta(P, h), \quad \alpha = \alpha(P, h, M, q).$$

Natomiast jednowymiarowy przepływ dwufazowy (mieszanina parowo-wodna) w strefie odparowania opisany jest następującym układem równań różniczkowych cząstkowych [32]:

$$\begin{aligned} \frac{\partial M}{\partial z} &= -\frac{\partial \rho}{\partial t} \\ \rho \frac{\partial h}{\partial t} + M \frac{\partial h}{\partial z} - \frac{\partial P}{\partial t} - \frac{M}{\rho} \frac{\partial P}{\partial z} &= q'_d - \frac{\partial}{\partial z} \left[M \left(\frac{w_2}{w} - 1 \right) (h - h') \right] \quad (127) \end{aligned}$$

$$\frac{\partial M}{\partial t} + 2 \frac{M}{\rho} \frac{\partial M}{\partial z} - \frac{M^2}{\rho^2} \frac{\partial \rho}{\partial z} + \rho g \cos \beta + 10^3 \frac{\partial P}{\partial z} + f \frac{M^2}{\rho} = 0$$

gdzie: $M = M(z, t) = \rho w$, $\rho = \rho(z, t) = \rho(P, h)$, $h = h(z, t)$, $P = P(z, t)$, $c = \frac{h-h'}{h_{21}}$,

$$\frac{w_2}{w} = \frac{1}{c + \frac{1-c}{S}} = f(M, P, h)$$

Warunki brzegowe dla rur ekranowych są następujące:

$$M(0, t) = M_w(t), h(0, t) = h_w(t), P(L, t) = P_{sep}(t).$$

Transport energii cieplnej wzdłuż promienia ścianki rury opisany jest następującą zależnością (przy założeniu, że właściwości materiału rury nie ulegają zmianie przy zmianach temperatury: $\lambda, \rho_r, c_r = const$):

$$\frac{\partial \Theta}{\partial t} = a \nabla^2 \Theta \quad (128)$$

W warunkach pracy parownika, gradienty temperatury w kierunku osiowym są pomijalnie małe w stosunku do gradientów w kierunku promieniowym. Na tej podstawie analizę przewodzenia ciepła w ściance rury parownika oparto na następującym równaniu różniczkowym:

$$\frac{\partial \Theta}{\partial t} = a \left(\frac{\partial^2 \Theta}{\partial r^2} + \frac{1}{r} \frac{\partial \Theta}{\partial r} \right) \quad (129)$$

warunki brzegowe dla tego równania są następujące:

$$\pi D \lambda \left(\frac{\partial \Theta}{\partial r} \right)_{r=\frac{D}{2}} = q \left[\frac{kW}{m} \right], \quad \pi d \lambda \left(\frac{\partial \Theta}{\partial r} \right)_{r=\frac{d}{2}} = \pi d \alpha (\Theta_w - \Theta) = q_d \left[\frac{kW}{m} \right]$$

gdzie: Θ_w - temperatura wewnętrznej powierzchni ścianki rury.

W przypadku parownika kotła BP-1150, poszczególne rury połączone są pletwą zapewniającą szczelność komory paleniskowej kotła o przekroju prostokątnym. Transport energii cieplnej wzdłuż pletwy jest opisany następującym równaniem przewodnictwa, z uwzględnieniem ciepła dopływającego z komory paleniskowej [32]:

$$\frac{1}{Podz p \rho_{st} c_{st}} q + a \frac{\partial^2 \Theta_p}{\partial x^2} = \frac{\partial \Theta_p}{\partial t} \quad (130)$$

z warunkami brzegowymi:

$$\left. \frac{\partial \Theta_p}{\partial t} \right|_{x=0} = 0, \quad \Theta_p \Big|_{x=L_p} = \Theta_p \Big|_{r=\frac{D}{2}}$$

gdzie: $a = \frac{\lambda}{\rho_{st} c_{st}} \left[\frac{m^2}{s} \right]$ - współczynnik wyrównywania temperatur,

$L_p = \frac{Podz-D}{2}$ [m] - połowa długości pletwy,

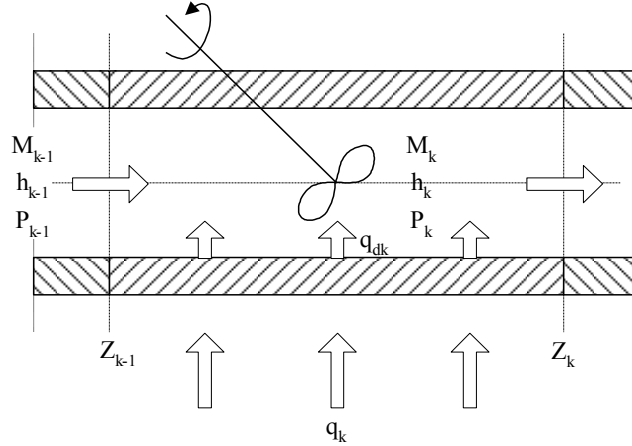
$Podz$ [m] - podziałka rozmieszczenia rur ekranowych,

p [m] - grubość pletwy.

Modele o parametrach skupionych rur ekranowych parownika

W celu uzyskania modelu o parametrach skupionych rur ekranowych parownika, podzielono zastępczą rurę parownika wzdłuż jej długości na pewną liczbę elementów

skończonych o wymiarach na tyle małych, aby można było założyć, że pojedynczy element stanowi układ dynamiczny o parametrach skupionych (opisany układem równań różniczkowych zwyczajnych) (rys. 78).



RYСУNEK 78. ELEMENT SKOŃCZONY MODELU O PARAMETRACH SKUPIONYCH

Na podstawie [32] równania bilansowe opisujące przepływ wody w k -tym elemencie skończonym wchodzącym w skład strefy dogrzewu przyjmują następującą postać:

$$\begin{aligned} \Delta Z_k \left(\frac{\partial \varrho}{\partial P} \right)_k \frac{dP_k}{dt} + \Delta Z_k \left(\frac{\partial \varrho}{\partial h} \right)_k \frac{dh_k}{dt} &= M_{k-1} - M_k \\ \Delta Z_k \varrho_k \frac{dh_k}{dt} - \Delta Z_k \frac{dP_k}{dt} &= M_{k-1} (h_{k-1} - h_k) - \frac{M_{k-1}}{\varrho_{k-1}} (P_{k-1} - P_k) + \frac{q_{dk}}{A} \Delta Z_k \\ \Delta Z_k \varrho_k \frac{dM_k}{dt} &= \frac{M_{k-1}^2}{\varrho_{k-1}} - \frac{M_k^2}{\varrho_k} - \Delta Z_k \varrho_k g \cos \beta - (P_{k+1} - P_k) 10^3 - \Delta Z_k f_k \frac{M_k^2}{\varrho_k} \end{aligned} \quad (131)$$

gdzie: $q_{dk} = \pi d \alpha_k (\theta_{WK} - \theta_K)$,

natomiast w k -tym elemencie skończonym strefy odparowania:

$$\begin{aligned} \Delta Z_k \left(\frac{\partial \varrho}{\partial P} \right)_k \frac{dP_k}{dt} + \Delta Z_k \left(\frac{\partial \varrho}{\partial h} \right)_k \frac{dh_k}{dt} &= M_{k-1} - M_k \\ \Delta Z_k \varrho_k \frac{dh_k}{dt} + \Delta Z_k \frac{dP_k}{dt} &= M_{k-1} (h_{k-1} - h_k) - \frac{M_{k-1}}{\varrho_{k-1}} (P_{k-1} - P_k) + \frac{q_{dk}}{A} \Delta Z_k + \dots \\ \dots + M_{k-1} \left(\left(\frac{w_2}{w} \right)_{k-1} - 1 \right) (h_{k-1} - h') &- M_k \left(\left(\frac{w_2}{w} \right)_k - 1 \right) (h_{k-1} - h') \\ \Delta Z_k \varrho_k \frac{dM_k}{dt} &= \frac{M_{k-1}^2}{\varrho_{k-1}} - \frac{M_k^2}{\varrho_k} - \Delta Z_k \varrho_k g \cos \beta - (P_{k+1} - P_k) 10^3 - \Delta Z_k f_k \frac{M_k^2}{\varrho_k} \end{aligned} \quad (132)$$

GDZIE: $q_{dk} = \pi d \alpha_k (\theta_{WK} - \theta_K)$, $\frac{w_2}{w} = \frac{1}{\frac{1-c}{S} + c}$, $S = f(P, h, M)$

Parametry czynnika roboczego w strefie dogrzewu jak również odparowania są funkcjami ciśnienia P oraz entalpii h : $\varrho(P, h)$, $\frac{\partial \varrho}{\partial P} = f(P, h)$, $\frac{\partial \varrho}{\partial h} = f(P, h)$:

$$\begin{aligned} \varrho &= f(P, h) = \frac{1}{(1-c)v' + cv''} = \frac{h_{21}}{v_{21}} \frac{1}{h - \frac{h'v'' - h''v'}{v_{21}}} \\ h_{21} &= h'' - h', \quad v_{21} = v'' - v' \\ c &= \frac{h-h'}{h_{21}}, \quad h' = h'(P), \quad h'' = h''(P) \end{aligned} \quad (133)$$

$$\left(\frac{\partial q}{\partial h}\right)_k = -\frac{q_k^2 v_{21}}{h_{21}}$$

$$\left(\frac{\partial q}{\partial P}\right)_k = -\frac{q_k v_{21}}{h_{21}} \frac{d}{dP} \left(\frac{h_{21}}{v_{21}}\right) + \frac{q_k^2 v_{21}}{h_{21}} \frac{d}{dP} \left(\frac{h' v'' - h'' v'}{v_{21}}\right)$$

$$W_2 = \frac{S_k}{1+c_k(S_k-1)} W$$

$$S_k = \frac{w_{2k}}{w_{1k}} = f(P_k, c_k, W_k)$$

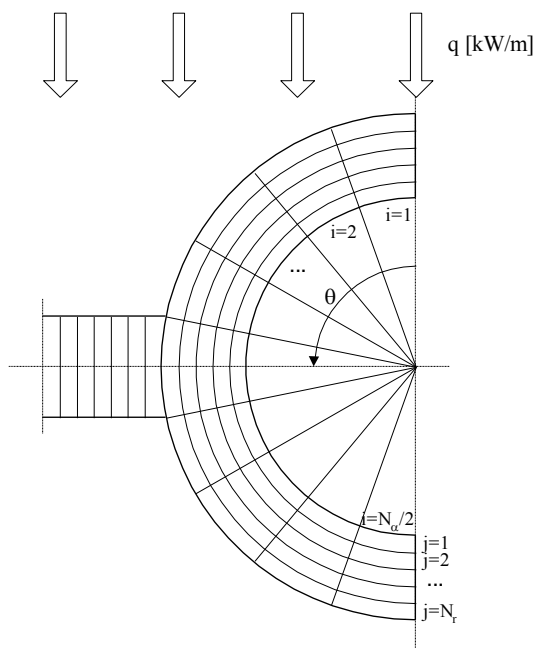
Trzy współrzędne stanu elementu skończonego są związane z czynnikiem roboczym: P_k , h_k , M_k , natomiast pozostałe współrzędne stanu są związane ze ścianką rury i płetwą łączącą.

Model o parametrach skupionych ścianki rury wraz z płetwą łączącą

Ekran komory paleniskowej kotła BP-1150 stanowią rury ekranowe parownika, połączone płetwą zapewniającą szczelność komory paleniskowej. Energia cieplna wytwarzana w komorze paleniskowej dopływa do ekranu od strony wewnętrznej, podczas gdy od strony zewnętrznej ekran jest izolowany termicznie. W znanych z literatury modelach matematycznych kotłów przyjmuje się, że ścianka rury ekranowej wraz z płetwą stanowią układ o parametrach skupionych, opisany równaniem różniczkowym zwyczajnym pierwszego rzędu (np.: [32]). Założenie takie zapewnia jednak niewielki zakres adekwatności modelu rur ekranowych parownika. Na podstawie [32] można stwierdzić, że zakres adekwatności wynosi w takim przypadku ok. 1 rad/s.

W celu poszerzenia zakresu adekwatności modelu rur ekranowych niezbędne staje się uwzględnienie rozłożenia parametrów modelu wzdłuż promienia oraz obwodu ścianki rury, a także wzdłuż płetwy łączącej.

Po dokonaniu dyskretyzacji ścianki rury oraz płetwy łączącej (rys. 79), równania przewodnictwa cieplnego przyjmują następującą postać [32]:



RYСУNEK 79. DYSKRETYZACJA ŚCIANKI RURY ORAZ PŁETWY ŁĄCZĄCEJ

Ścianka rury:

$$\begin{aligned}
\frac{d \Theta_k^{i,j}}{dt} &= \frac{4 \lambda_{tw}}{\rho_{tw} c_{tw} \Delta} \left[\frac{(d+2j\Delta)(\Theta_k^{i,j+1} - \Theta_k^{i,j}) - (d+2(j-1)\Delta)(\Theta_k^{i,j} - \Theta_k^{i,j-1})}{[(d+2j\Delta)^2 - (d+2(j-1)\Delta)^2]} + \right. \\
&\quad \left. + \frac{4 N_\alpha^2 \Delta \lambda_{tw}}{\pi^2 \rho_{tw} c_{tw} [d+(2j-1)\Delta]} \frac{\Theta_k^{i-1,j} - 2\Theta_k^{i,j} + \Theta_k^{i+1,j}}{[(d+2j\Delta)^2 - (d+2(j-1)\Delta)^2]} \right] \\
\frac{d \Theta_k^{i,1}}{dt} &= - \frac{4(d+2\Delta)\lambda_{tw}}{[(d+2\Delta)^2 - d^2] \rho_{tw} c_{tw} \Delta} (\Theta_k^{i,2} - \Theta_k^{i,1}) - \frac{4d\alpha_k}{[(d+2\Delta)^2 - d^2] \rho_{tw} c_{tw}} (\Theta_k^{i,1} - \vartheta_k) + \\
&\quad + \frac{4 N_\alpha^2 \lambda_{tw} \Delta}{\pi^2 (d+\Delta) [(d+2\Delta)^2 - d^2] \rho_{tw} c_{tw}} (\Theta_k^{i-1,1} - 2\Theta_k^{i,1} + \Theta_k^{i+1,1}) \\
\frac{d \Theta_k^{i,N_{tw}}}{dt} &= \frac{4 N_\alpha}{\pi [D^2 - (D-2\Delta)^2] \rho_{tw} c_{tw}} \frac{D}{D+L_{bl}} \frac{1}{\frac{N_\alpha}{2} - 1} q + \\
&\quad + \frac{4 \lambda_{tw} (D-2\Delta)}{[D^2 - (D-2\Delta)^2] \rho_{tw} c_{tw} \Delta} (\Theta_k^{i,N_{tw}-1} - \Theta_k^{i,N_{tw}}) + \\
&\quad + \frac{4 N_\alpha^2 \Delta \lambda_{tw}}{\pi^2 (D-\Delta) [D^2 - (D-2\Delta)^2] \rho_{tw} c_{tw}} [\Theta_k^{i-1,N_{tw}} - 2\Theta_k^{i,N_{tw}} + \Theta_k^{i+1,N_{tw}}]
\end{aligned} \tag{134}$$

gdzie:

$\Delta = \frac{D-d}{2 N_{tw}}$, N_{tw} – liczba warstw modelu dyskretnego ścianki rury, N_α – liczba elementów

wzdłuż obwodu rury, L_{bl} – długość pletwy łączącej.

Pletwa łącząca:

$$\begin{aligned}
\frac{d \Theta_{blk}^l}{dt} &= \frac{\lambda_{bl}}{\Delta_{bl}^2 \rho_{bl} c_{bl}} [\Theta_{blk}^{l-1} - 2\Theta_{blk}^l + \Theta_{blk}^{l+1}] + \frac{L_{bl}}{D+L_{bl}} \frac{1}{N_{bl} \delta_{bl} \Delta_{bl} \rho_{bl} c_{bl}} q_k \\
\frac{d \Theta_{blk}^1}{dt} &= \frac{\lambda_{bl}}{\Delta_{bl}^2 \rho_{bl} c_{bl}} (\Theta_{blk}^{i_{bl}, N_{tw}} - \Theta_{blk}^1) + \frac{\lambda_{bl}}{\Delta_{bl}^2 \rho_{bl} c_{bl}} (\Theta_{blk}^2 - \Theta_{blk}^1) + \\
&\quad + \frac{L_{bl}}{D+L_{bl}} \frac{1}{N_{bl} \delta_{bl} \Delta_{bl} \rho_{bl} c_{bl}} q_k \\
\frac{d \Theta_{blk}^{N_{bl}}}{dt} &= \frac{\lambda_{bl}}{\Delta_{bl}^2 \rho_{bl} c_{bl}} (\Theta_{blk}^{N_{bl}-1} - \Theta_{blk}^{N_{bl}}) + \frac{L_{bl}}{D+L_{bl}} \frac{1}{N_{bl} \delta_{bl} \Delta_{bl} \rho_{bl} c_{bl}} q_k
\end{aligned} \tag{135}$$

gdzie: δ_{bl} - grubość pletwy łączącej, $\Delta_{bl} = \frac{L_{bl}}{N_{bl}}$.

W modelu rur ekranowych założono, że $N_{tw}=5$, $N_{bl}=7$, $N_\alpha=18$. W wyniku takiej dyskretyzacji model matematyczny elementu skończonego zawiera 55 zmiennych stanu (trzy związane ze stanem czynnika roboczego – P_k , h_k , M_k oraz 52 zmienne związane z temperaturą ścianki rury oraz pletwy łączącej – 45 zmiennych stanu dla temperatury ścianki rury oraz 7 zmiennych stanu dla temperatury pletwy łączącej).

6.3.2. Separator

Rola separatora w kotłach przepływowych firmy SULZER polega na oddzielaniu wody z mieszaniny parowo-wodnej opuszczającej rury ekranowe parownika. Separator stanowi wydłużony, pionowy zbiornik cylindryczny o długości $L_{sep} \cong 34 \text{ m}$. i średnicy wewnętrznej $d=0.8 \text{ m}$, oraz średnicy zewnętrznej $D=0.94 \text{ m}$. Do górnej części separatora doprowadzana jest mieszanina parowo-wodna z prędkością ok. 16 m/s, stycznie usytuowanymi króćcami. Dolnymi króćcami, również usytuowanymi stycznie, jest odprowadzana woda do mieszalnika z prędkością ok. 2.5 m/s. Mechanizm rozdziału faz w polu sił odśrodkowych występujących w separatorze jest bardzo złożonym procesem, zależnym od szeregu wielkości fizycznych oraz parametrów konstrukcyjnych i eksploatacyjnych [32].

Separator jako obiekt sterowania jest układem o czterech wielkościach wejściowych: M_p^* , M_{sep}^* , h_{sep}^* , M_{kond}^* i dwóch wielkościach wyjściowych: P_{sep} , H_{sep} . W pracy zastosowano model separatora o parametrach skupionych, wyznaczony na podstawie równań bilansu masy oraz energii wewnętrznej czynnika roboczego, a także bilansu energii cieplnej w grubościennym płaszczu zewnętrznym. Równanie bilansu masy w separatorze ma następującą postać:

$$M_{sep}^* - M_{p_{sep}}^* - M_{kond}^* = \frac{d}{dt} [V' \rho' + V'' \rho'']$$

gdzie

$$V'' = V - V' \quad \text{– objętość pary w separatorze}$$

$$V' = AH_{sep} \quad \text{– objętość wody w separatorze.}$$

Na tej podstawie uzyskuje się następujące równanie bilansu masy w separatorze:

$$M_{sep}^* - M_{p_{sep}}^* - M_{kond}^* =$$

$$A(\rho' - \rho'') \frac{dH_{sep}}{dt} + \left[AH_{sep} \frac{d\rho'}{dP} + (V - AH_{sep}) \frac{d\rho''}{dP} \right] \frac{dP_{sep}}{dt}. \quad (137)$$

Równanie bilansu energii cieplnej dla separatora można zapisać następująco:

$$M_{sep}^* h_{sep}^* - M_p^* h'' - M_{kond}^* h' - L_{sep} q_d =$$

$$= \left[AH_{sep} \frac{d(\rho' h')}{dP} + (V - AH_{sep}) \frac{d(\rho'' h'')}{dP} \right] \frac{dP_{sep}}{dt} \quad (138)$$

$$+ A[\rho' h' - \rho'' h''] \frac{dH_{sep}}{dt},$$

gdzie: $q_d = \pi d \alpha (\vartheta - \Theta_w)$ - strumień cieplny odniesiony do 1 m . długości separatora, wymieniany między czynnikiem roboczym w separatorze a płaszczem separatora,

α - współczynnik wnikania ciepła między czynnikiem roboczym w separatorze a ścianą separatora,

Θ_w - temperatura wewnętrznej powierzchni ściany separatora.

Na podstawie powyższych równań bilansowych uzyskuje się następujące równania stanu opisujące bilans masy oraz energii cieplnej czynnika roboczego w separatorze:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} \frac{dP_{sep}}{dt} \\ \frac{dH_{sep}}{dt} \end{bmatrix} = \begin{bmatrix} M_{sep}^* - M_p^* - M_{kond}^* \\ M_{sep}^* h_{sep}^* - M_p^* h'' - M_{kond}^* h' - \pi d \alpha L_{sep} (\vartheta - \Theta_1) \end{bmatrix}, \quad (139)$$

gdzie:

$$\begin{aligned} a &= A_z H_{sep} \frac{d\rho'}{dP} + (V - A_z H_{sep}) \frac{d\rho''}{dP}, \\ b &= A_z (\rho' - \rho''), \\ c &= A_z H_{sep} \frac{d(\rho' h')}{dP} + (V - A_z H_{sep}) \frac{d(\rho'' h'')}{dP}, \\ d &= A_z (\rho' h' - \rho'' h''). \end{aligned}$$

Przy założeniu, że temperatura płaszczu separatora jest jednakowa wzdłuż obwodu, w wyniku dyskretyzacji równania bilansu energii cieplnej w płaszczu separatora przyjmują następującą postać:

$$\frac{d}{dt} \begin{bmatrix} \Theta_1 \\ \dots \\ \Theta_k \\ \dots \\ \Theta_N \end{bmatrix} = \begin{bmatrix} 4 \frac{d\alpha\Delta(\vartheta - \Theta_1) - (d+2\Delta)\lambda(\Theta_1 - \Theta_2)}{[(d+2\Delta)^2 - d^2] \rho_{st} c_{st} \Delta} \\ \dots \\ 4 \frac{[d+2(k-1)\Delta]\lambda(\Theta_{k-1} - \Theta_k) - (d+2k\Delta)\lambda(\Theta_k - \Theta_{k+1})}{\{(d+2k\Delta)^2 - [d+2(k-1)\Delta]^2\} \rho_{st} c_{st} \Delta} \\ \dots \\ 4 \frac{[d+2(N-1)\Delta]\lambda(\Theta_{N-1} - \Theta_N)}{[D^2 - (D-2\Delta)^2] \rho_{st} c_{st} \Delta} \end{bmatrix}, \quad (140)$$

Modele matematyczne pozostałych podsystemów parownika kotła BP-1150 przedstawiono w pracy [32].

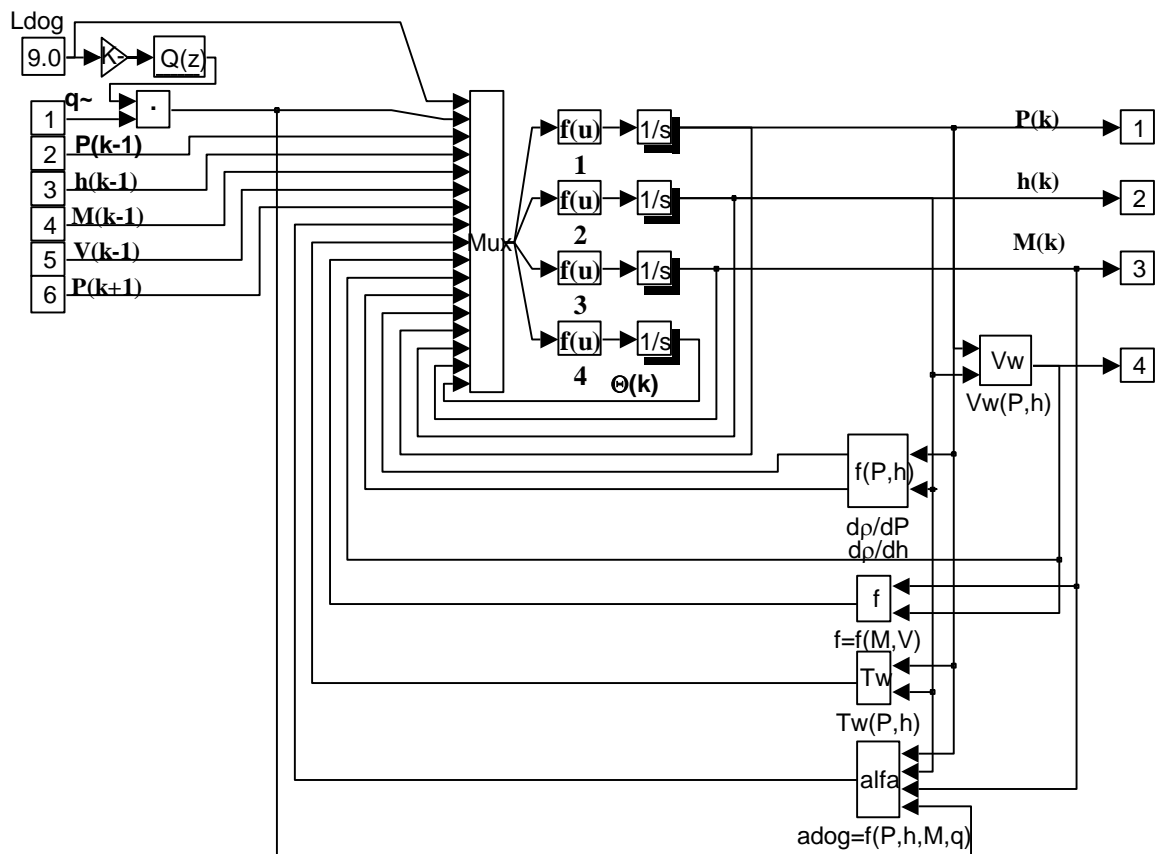
6.4. Modele zlinearyzowane

W pracy [32] nieliniowe modele matematyczne poszczególnych podsystemów parownika kotła energetycznego BP-1150 zrealizowano w środowisku MATLAB/Simulink. Na rys. 80 przedstawiono model matematyczny elementu skończonego na przykładzie strefy dogrzewu rur ekranowych parownika. Połączenie modeli elementów skończonych pozwala zbudować model strefy dogrzewu o praktycznie dowolnym rozmiarze.

Dysponując nieliniowymi równaniami różniczkowymi opisującymi procesy zachodzące w parowniku kotła energetycznego należy wyznaczyć wartości poszczególnych wielkości w stanie ustalonym. Jest to zadanie złożone, gdyż wymaga rozwiązania układu równań, uzyskanego po wyzerowaniu pochodnych względem czasu. Wymaga to złożonego

procesu iteracyjnego, w którym należy wyznaczyć rozkłady ciśnienia, entalpii, temperatury czynnika roboczego wzdłuż rur ekranowych, jak również rozkłady temperatur wzdłuż promienia ścianki rury oraz wzdłuż płetwy łączącej. Operacje te trzeba wykonać dla różnych wartości obciążenia bloku energetycznego.

Dysponując punktami pracy poszczególnych elementów skończonych modelu parownika można wykonać operację linearyzacji modeli podsystemów parownika w danym punkcie pracy. Operację tą można wykonać analitycznie, jednak wymagałoby to zbyt długiego czasu. W pracy do tej operacji wykorzystano MATLAB, linearyzując modele poszczególnych elementów skończonych z zastosowaniem funkcji `linmod`, a następnie łącząc zlinearyzowane modele w stworzenia modeli podsystemów parownika.



RYSUNEK 80. MODEL ELEMENTU SKOŃCZONEGO STREFY DOGRZEWU W MATLAB/SIMULINK

6.5. Uwagi

Modele matematyczne dla celów sterowania kotłów energetycznych rozwijają się od kilkudziesięciu lat. Do klasycznych pozycji w tym zakresie należy zaliczyć książkę P. Profosa z 1962 roku: *Die Regelung von Dampfanlagen*, Springer, przedstawiającą modele matematyczne oraz analizę właściwości dynamicznych kotłów energetycznych. Analizując bieżącą literaturę naukową, dotyczącą modelowania dynamiki kotłów energetycznych, należy stwierdzić, że zagadnienie to jest nadal bardzo aktualne [145] [146] [147] [148], szczególnie dla nowych konstrukcji kotłowych.

Modele matematyczne stają się coraz bardziej złożone, szczególnie ze względu na opis procesów o parametrach rozłożonych z zastosowaniem metody elementów skończonych. Mamy do czynienia z szybkim rozwojem oprogramowania do modelowania procesów

termodynamicznych oraz ze wzrostem mocy obliczeniowych komputerów, co umożliwia budowę i analizę modeli matematycznych o coraz większej złożoności.

Jednocześnie mamy do czynienia z rozwojem algorytmów sterowania, wymagających dokładnych i adekwatnych w szerokim zakresie zmian wielkości, jak również w szerokim zakresie częstotliwości, modeli skonstruowanych dla celów sterowania. Współczesne komputerowe systemy sterowania pozwalają na realizację niemal dowolnie złożonych algorytmów sterowania, pod warunkiem, że dysponujemy adekwatnym modelem obiektu sterowania.

W pracy doktorskiej głównie skupiono się na problemie redukcji rzędu modeli złożonych obiektów sterowania. W tym celu opracowano i zaimplementowano algorytmy redukcji modeli zlinearyzowanych wysokiego rzędu. Modele matematyczne używane w pracy zostały zaczerpnięte z prac prowadzonych przez W. Stanisławskiego [32].

7. Zredukowane modele podsystemów kotła energetycznego BP-1150

Modele matematyczne podsystemów parownika kotła BP-1150, uzyskane metodą elementów skończonych, zawierają wiele tysięcy zmiennych stanu. Ze względu na wysoką złożoność obliczeniową operacji redukcji rzędu modeli z zastosowaniem SVD oraz metody częstotliwościowych funkcji wagowych konieczne jest zastosowanie obliczeń równoległych. W tym celu modele poszczególnych podsystemów parownika zostają poddane dekompozycji (jednopoziomowej lub dwupoziomowej). Uzyskane modele zdekomponowane zostają niezależnie od siebie poddane operacji redukcji, a następnie scalane są ze sobą z uwzględnieniem interfejsów wyznaczonych na etapie dekompozycji. Ponieważ zredukowany model parownika, po wykonaniu operacji scalania zdekomponowanych modeli zredukowanych, posiada nadal wysoki rząd (ponad 1000 zmiennych stanu), konieczna jest ponowna operacja redukcji rzędu, aby uzyskany zredukowany model parownika mógł być użyty do projektowania układów sterowania.

7.1. Wybór metod dekompozycji dla modeli podsystemów parownika

Na podstawie rozważań przeprowadzonych w rozdziałach 3 i 4, wyszczególniono 14 metod dekompozycji jednopoziomowej oraz 78 metod dekompozycji dwupoziomowej. Szereg z opracowanych metod dekompozycji, przy zastosowaniu do modeli podsystemów parownika, nie spełnia oczekiwań. Pod uwagę brano przede wszystkim:

- stabilność modelu zredukowanego, uzyskanego na podstawie scalania zdekomponowanych modeli zredukowanych,
- błąd aproksymacji charakterystyk częstotliwościowych dla danego przedziału częstotliwości,
- czas trwania obliczeń.

Określenie właściwości wszystkich wyszczególnionych metod dekompozycji dla modeli podsystemów parownika wysokiego rzędu jest praktycznie zadaniem niewykonalnym przy korzystaniu z dostępnego sprzętu komputerowego. Aby poddać analizie wszystkie wyszczególnione metody dekompozycji, obliczenia przeprowadzono z zastosowaniem specjalnie opracowanych modeli podsystemów parownika o niskim rzędzie. Dla dekompozycji jednopoziomowej, obliczenia przeprowadzono z zastosowaniem modelu strefy dogrzewu rur ekranowych o rzędzie 300 (20 elementów skończonych piętnastego rzędu), natomiast dla dekompozycji dwupoziomowych – z zastosowaniem modelu rzędu 600 (40 elementów skończonych piętnastego rzędu). Uzyskane wyniki pozwalają ocenić właściwości poszczególnych metod dekompozycji w zastosowaniu do modeli podsystemów parownika, z użyciem redukcji metodą FW. Z analizy procesów fizycznych zachodzących w parowniku przepływowym oraz założeń upraszczających, które legły u podstaw opracowania modeli matematycznych, zakres adekwatności modeli podsystemów rur ekranowych (strefa dogrzewu, strefa odparowania I, II, III) można ocenić na około $\omega_{max} = 20 \text{ rad/s}$ [32]. Dla modeli matematycznych dla celów sterowania dolną częstotliwością zakresu adekwatności jest $\omega_{min} = 0 \text{ rad/s}$ (zgodność z obiektem dla stanu ustalonego). Dla logarytmicznych charakterystyk częstotliwościowych przyjęto $\omega_{min} = 10^{-3} \text{ rad/s}$.

Dla wszystkich modeli zredukowanych podsystemów rur ekranowych parownika błąd aproksymacji charakterystyk częstotliwościowych jest wyznaczany dla przedziału częstotliwości $\langle 10^{-3} - 20 \text{ rad/s} \rangle$. Jednak dla zapewnienia stabilności modeli

zredukowanych, uzyskanych na podstawie scalania zdekomponowanych modeli zredukowanych, konieczna jest aproksymacja charakterystyk częstotliwościowych w zakresie od ok. 1000 rad/s . Wynika to z faktu, że w powiązaniach między zdekomponowanymi modelami funkcjonuje silne sprzężenie zwrotne, oddziaływujące na wysokich częstotliwościach.

Aby zapewnić niskie błędy aproksymacji charakterystyk częstotliwościowych dla wysokich częstotliwości modeli zdekomponowanych, modele zredukowane muszą być wysokiego rzędu. Pociąga to za sobą konieczność ponownej redukcji modelu parownika, uzyskanego na podstawie połączonych zredukowanych modeli podsystemów, w celu uzyskania modelu zredukowanego parownika o niskim rzędzie, który będzie mógł być zastosowany do projektowania układów sterowania.

Zakres adekwatności modelu parownika kotła BP-1150 wynosi ok. $0,5 \text{ rad/s}$, co w pełni wystarcza do prawidłowego zaprojektowania układu sterowania.

7.1.1. Analiza błędu oraz stabilności zredukowanych modeli dla dekompozycji jednopoziomowej

Do analizy metod dekompozycji dla rozważanych modeli podsystemów rur ekranowych parownika kotła energetycznego BP-1150 przyjęto następujące założenia:

- rozmiar modelu wynosi 300 zmiennych stanu,
- limit redukcji algorytmu ewolucyjnego dla metody FW [15] - 25,
- limit redukcji algorytmu poszukiwania minimum lokalnego [15] - 10,
- zakres aproksymacji charakterystyk częstotliwościowych dla metody FW jest zmieniany od $\langle 10^{-3} - 10^{1,3} \text{ rad/s} \rangle$ do $\langle 10^{-3} - 10^3 \text{ rad/s} \rangle$ gdzie kolejne wartości prawego brzegu przedziału aproksymacji wynoszą $10^{x+0,1}$ (x – poprzednia wartość wykładnika). Daje to w sumie 18 różnych zakresów aproksymacji dla danego rzędu modelu zredukowanego,
- rząd modelu wynikowego (lub wolnozmiennnej części modelu w przypadku dekompozycji slow-fast) zmienia się począwszy od 20 (6,7% zmiennych stanu modelu oryginalnego) do 100 (33,3% zmiennych stanu modelu oryginalnego) z krokiem 5. Dla dekompozycji slow-fast założono dodatkowo, że rozmiar szybkozmiennnej części modelu będzie równy 40 (modele wynikowe rzędu od 60 do 140). Daje to w sumie 17 różnych rzędów modeli dla danego zakresu adekwatności,
- za miarę błędu modelu wynikowego przyjęto średniokwadratowy błąd względny aproksymacji charakterystyk częstotliwościowych, wyrażony zależnością [15]:

$$\rho = \sqrt{\frac{1}{pmN} \sum_{i=1}^N \sum_{x=1}^p \sum_{y=1}^m \left(\frac{|G_{(x,y)}(j\omega_i) - G_{r(x,y)}(j\omega_i)|}{|G_{(x,y)}(j\omega_i)|} \right)^2} \quad (141)$$

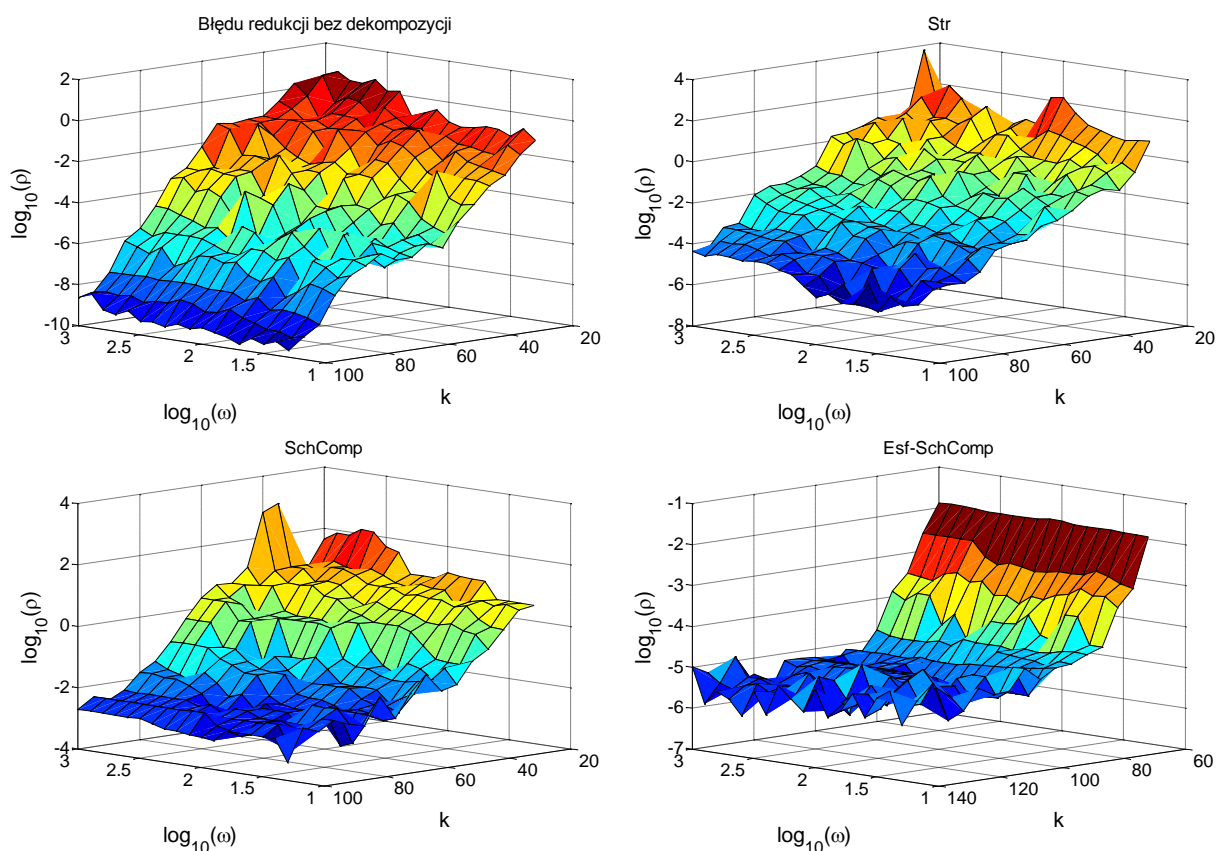
gdzie:

p , m - liczba wejść i wyjść modelu, N - liczba aproksymowanych punktów dla dziedziny częstotliwościowej w przedziale $\langle 10^{-3} - 20 \text{ rad/s} \rangle$

- Za punkt odniesienia przyjęto wyniki uzyskane z redukcji modelu oryginalnego bez użycia dekompozycji.

Na rys. 81 przedstawiono przykładowe błędy aproksymacji modeli zredukowanych w funkcji rzędu modelu zredukowanego oraz zakresu aproksymacji charakterystyk częstotliwościowych dla wybranych metod dekompozycji modelu pierwotnego. Komplet wyników przedstawiono

w **dodatku G**. W tabeli 2 zostały zebrane sumaryczne wyniki dla przebadanych metod dekompozycji jednopoziomowej. Prezentowane są informacje na temat minimalnego błędu, który został otrzymany spośród 306 wyznaczonych modeli dla danej dekompozycji oraz procentu modeli stabilnych uzyskanych w wyniku dekompozycji i redukcji.



RYСУNEK 81. PORÓWNIANIE BŁĘDU MODELI ZREDUKOWANYCH Z WYKORZYSTANIEM RÓŻNYCH TYPÓW DEKOMPOZYCJI

Redukcja modelu bez stosowania dekompozycji charakteryzuje się najwyższą dokładnością (średniokwadratowy błąd względny charakterystyk częstotliwościowych w zakresie $\langle 10^{-3} - 20 \text{ rad/s} \rangle$ wynosi ok. 10^{-9}). Tak niski błąd aproksymacji jest nieosiągalny dla redukcji z zastosowaniem dekompozycji. Nieco wyższy błąd aproksymacji uzyskano dla dekompozycji Schur-Sylwester (ok. 10^{-8}), strukturalnej pełnej (ok. 10^{-6}), Epsilon slow-fast połączonej z Schur-Complement (ok. 10^{-6}), Epsilon slow-fast połączonej z Schur-Sylwester (ok. 10^{-4}). Metody dekompozycji Schura oraz Schur-Complement dają błąd aproksymacji ok. 10^{-3} . Pozostałe metody charakteryzują się wysokim błędem aproksymacji i zastosowanie ich do redukcji podsystemów rur ekranowych jest niecelowe.

Drugim problemem z którym spotykamy się podczas redukcji modelu z zastosowaniem dekompozycji jest możliwość niestabilności modelu zredukowanego, otrzymanego na podstawie połączenia zdekomponowanych modeli zredukowanych. Niestabilność wynika z faktu, że modyfikacji ulegają zarówno modele podsystemów (poprzez proces redukcji rzędu) jak i interfejs je łączący (w wyniku modyfikacji macierzami transformacji wyznaczonymi w procesie redukcji podsystemów). W przypadku, gdy zdekomponowane modele są połączone silnymi sprzężeniami zwrotnymi (a tak jest w tym przypadku), model zredukowany może okazać się niestabilny (zdekomponowane modele zredukowane posiadają nieco zmodyfikowane charakterystyki częstotliwościowe). W celu uniknięcia niestabilności modelu zredukowanego, uzyskanego na podstawie połączenia

zdekomponowanych modeli zredukowanych, należy zastosować wyższą częstotliwość aproksymacji charakterystyk częstotliwościowych i/lub wyższy rząd modelu zredukowanego.

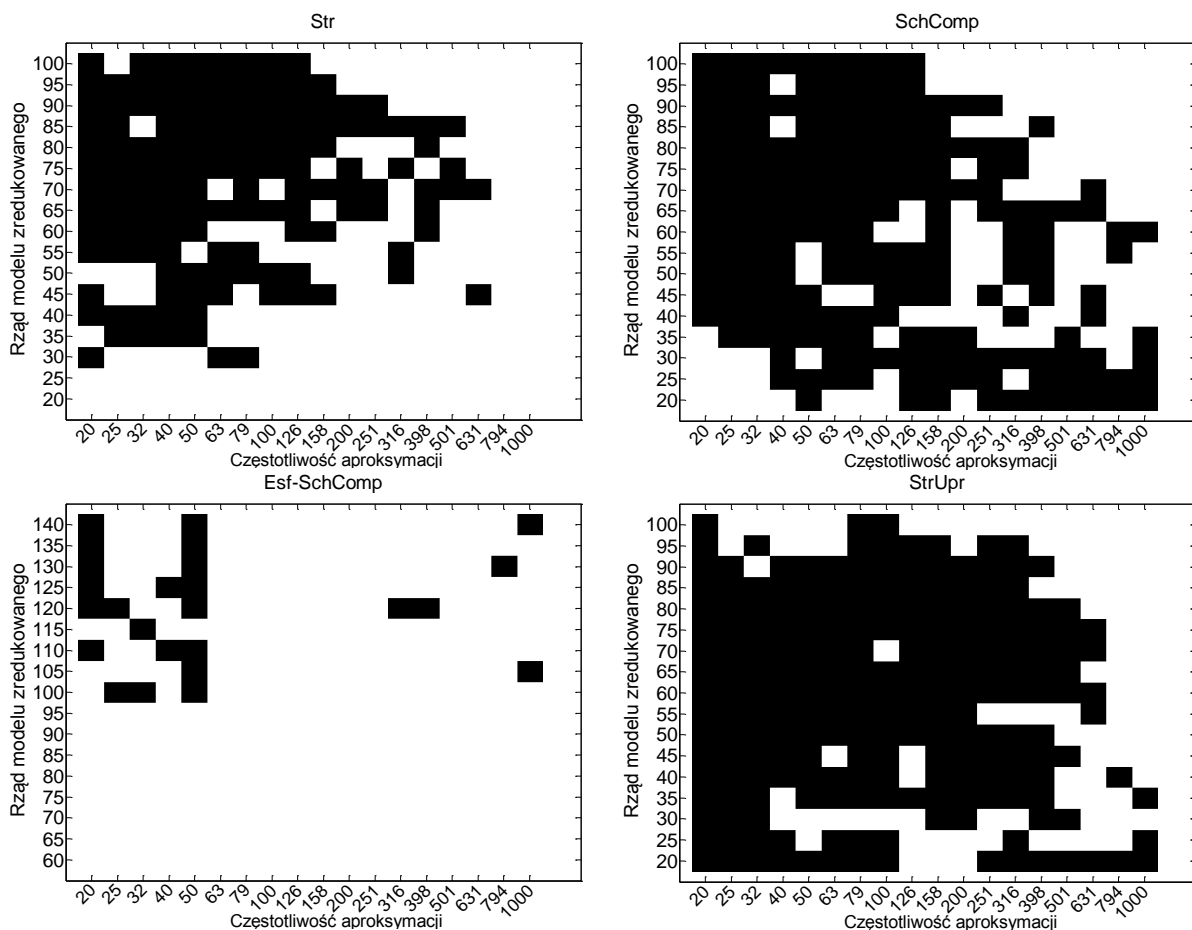
Na rys. 82 przedstawiono mapy stabilności modeli zredukowanych, uzyskane na podstawie połączenia zdekomponowanych modeli zredukowanych przy zastosowaniu wybranych metod dekompozycji.

TABELA 2. SUMARYCZNE WYNIKI DLA PRZEBADANYCH METOD DEKOMPOZYCJI JEDNOPOZIOMOWEJ W POŁĄCZENIU Z REDUKCJĄ

dekompozycja	minimalny błąd	% stabilnych modeli
<i>Str</i>	7.5050e-007	56,21%
<i>SchSylv</i>	3.1232e-009	100%
<i>SchComp</i>	3.6848e-004	36,6%
<i>StrUpr</i>	0.1831	30,72%
<i>SchCompZm</i>	0.0099	16,99%
<i>Esf-Str</i>	-	-
<i>Esf-StrUpr</i>	0.5503	8,17%
<i>Esf-MLS</i>	0.2725	100%
<i>Esf-SchComp</i>	9.6397e-007	91,83%
<i>Sch</i>	1.2718e-004	100%
<i>Esf-SchSylv</i>	2.7292e-005	100%
<i>Esf-Sch</i>	0.0011	100%
<i>Esf-SchCompZm</i>	0.0056	100%
<i>MATLAB slowfast</i>	0.0090	100%

Jedynie metody dekompozycji eliminujące sprzężenia zwrotne między zdekomponowanymi podsystemami – *SchSylv*, *Esf-MLS*, *Sch*, *Esf-SchSylv*, *Esf-Sch*, *MATLAB slowfast*, oraz metoda *Esf-SchCompZm* zapewniają stabilność modelu zredukowanego.

Istotne przyspieszenie obliczeń związanych z redukcją uzyskuje się w przypadku, gdy zdekomponowane podsystemy mają zbliżony rozmiar. Ma to miejsce we wszystkich metodach dekompozycji z wyjątkiem dekompozycji Slow-fast. W tej metodzie wydziela się podsystem charakteryzujący się dużymi wartościami własnymi. W analizowanych modelach jedynie dwa równania stanu w każdym elemencie skończonym (15 zmiennych stanu) opisują szybkie procesy dynamiczne związane z przemieszczaniem się zmian ciśnienia wzdłuż rur ekranowych parownika, z którymi są związane zespolone wartości własne o dużym module. Wynika z tego, że podczas zastosowania dekompozycji slow-fast uzyskuje się dwa podsystemy o znacznie różniących się rozmiarach, co istotnie wydłuża proces redukcji modeli.



RYSUNEK 82. MAPY STABILNOŚCI MODELI ZREDUKOWANYCH (kolor biały – model stabilny, kolor czarny – model niestabilny)

7.1.2. Analiza błędów oraz stabilności zredukowanych modeli dla dekompozycji dwupoziomowej

Do analizy metod dekompozycji dwupoziomowej dla modeli podsystemów parownika, przyjęto analogiczne kryteria jak w przypadku dekompozycji jednopoziomowej. Obliczenia zostały przeprowadzone na modelu rzędu 600. Do redukcji modeli podsystemów parownika zastosowano 78 metod dekompozycji przedstawionych w rozdziale 4 (Metoda dekompozycji na pierwszym poziomie/Metoda dekompozycji na drugim poziomie) (rys.83).

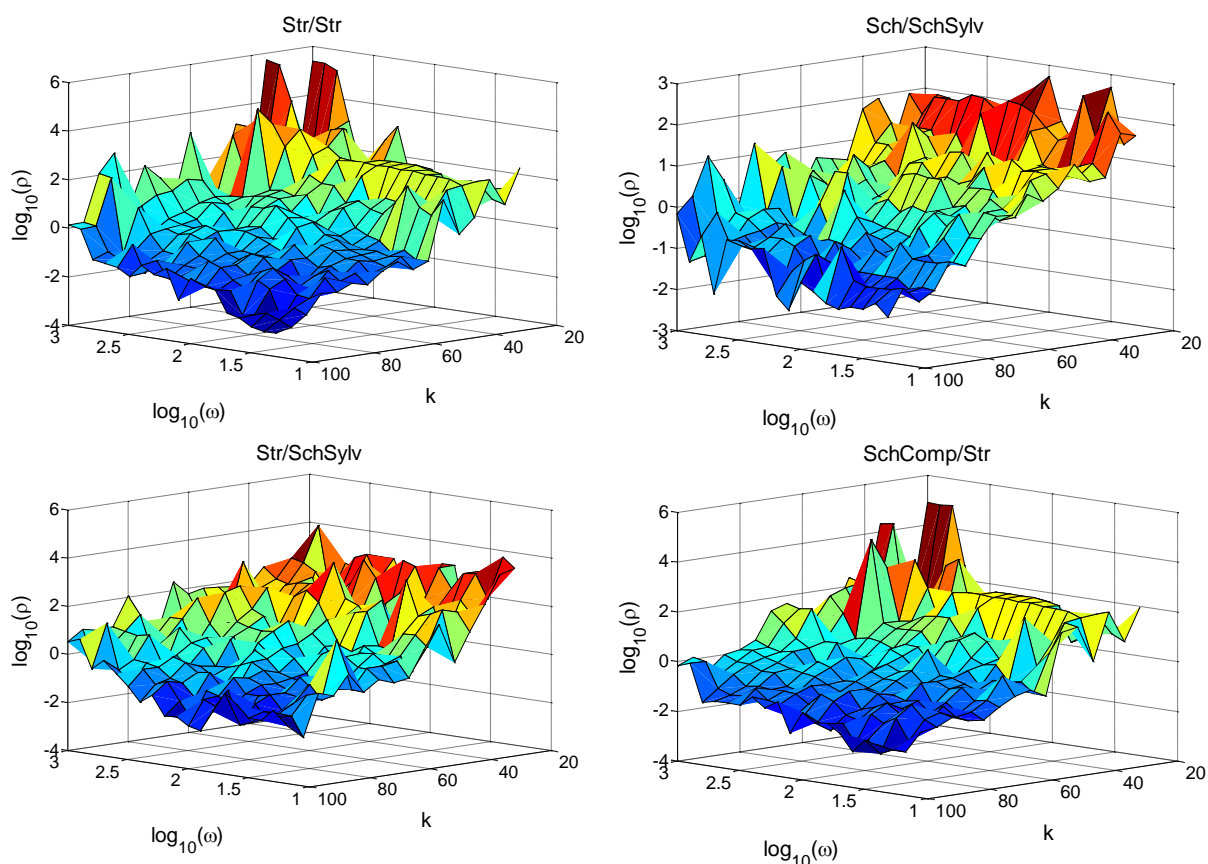
Na rys. 84 przedstawiono przykładowe błędy aproksymacji modeli zredukowanych w funkcji rzędu modelu zredukowanego oraz zakresu aproksymacji charakterystyk częstotliwościowych dla wybranych metod dekompozycji dwupoziomowej modelu pierwotnego. Komplet wyników przedstawiono w **dodatku H**. Sumaryczne wyniki określające właściwości danej kombinacji dekompozycji w połączeniu z redukcją FW zostały zebrane w tabeli 3. Pierwsza kolumna tabeli 3 określa metody dekompozycji użyte na pierwszym poziomie dekompozycji, pierwszy wiersz natomiast metody wykorzystane na drugim poziomie dekompozycji.

Biorąc pod uwagę błąd aproksymacji modeli zredukowanych, do najkorzystniejszych metod dekompozycji dwupoziomowej należy zaliczyć te, które na pierwszym poziomie wykorzystują dekompozycję *Esf-SchSylv* lub *Esf-SchComp*. Stosując te metody, w pierwszej kolejności w modelu wydziela się część szybką i wolną (permutacja wierszy i kolumn modelu ze względu na wartości własne i zastosowanie dekompozycji Schur-

Sylvester lub Schur-Complement). Uzyskane modele poddawane są następnie jednej z dekompozycji: *SchComp*, *SchCompZm*, *SchSylv*, *StrUpr*.

1. <i>Str/Str</i> ,	27. <i>SchComp/Sch</i> ,	53. <i>Esf-StrUpr/SchComp</i> ,
2. <i>Str/StrUpr</i> ,	28. <i>SchComp/SchSylv</i> ,	54. <i>Esf-StrUpr/SchCompZm</i> ,
3. <i>Str/Sch</i>	29. <i>SchComp/SchComp</i> ,	55. <i>Esf-Str/Str</i> ,
4. <i>Str/SchSylv</i> ,	30. <i>SchComp /SchCompZm</i> ,	56. <i>Esf-Str/StrUpr</i> ,
5. <i>Str/SchComp</i> ,	31. <i>SchCompZm/Str</i> ,	57. <i>Esf-Str/Sch</i>
6. <i>Str/SchCompZm</i> ,	32. <i>SchCompZm/StrUpr</i> ,	58. <i>Esf-Str/SchSylv</i> ,
7. <i>StrUpr/Str</i> ,	33. <i>SchCompZm/Sch</i>	59. <i>Esf-Str/SchComp</i> ,
8. <i>StrUpr/StrUpr</i> ,	34. <i>SchCompZm/SchSylv</i> ,	60. <i>Esf-Str/SchCompZm</i> ,
9. <i>StrUpr/Sch</i> ,	35. <i>SchCompZm/SchComp</i> ,	61. <i>Esf-Sch/Str</i> ,
10. <i>StrUpr/SchSylv</i> ,	36. <i>SchCompZm /SchCompZm</i> ,	62. <i>Esf-Sch/StrUpr</i> ,
11. <i>StrUpr/SchComp</i> ,	37. <i>Esf-SchComp/Str</i> ,	63. <i>Esf-Sch/Sch</i>
12. <i>StrUpr/SchCompZm</i> ,	38. <i>Esf-SchComp/StrUpr</i> ,	64. <i>Esf-Sch/SchSylv</i> ,
13. <i>Sch/Str</i> ,	39. <i>Esf-SchComp/Sch</i>	65. <i>Esf-Sch/SchComp</i> ,
14. <i>Sch/StrUpr</i> ,	40. <i>Esf-SchComp/SchSylv</i> ,	66. <i>Esf-Sch/SchCompZm</i> ,
15. <i>Sch/Sch</i> ,	41. <i>Esf-SchComp/SchComp</i> ,	67. <i>Esf-MLS/Str</i> ,
16. <i>Sch/SchSylv</i> ,	42. <i>Esf-SchComp/SchCompZm</i> ,	68. <i>Esf-MLS/StrUpr</i> ,
17. <i>Sch/SchComp</i> ,	43. <i>Esf-SchSylv/Str</i> ,	69. <i>Esf-MLS/Sch</i>
18. <i>Sch/SchCompZm</i> ,	44. <i>Esf-SchSylv/StrUpr</i> ,	70. <i>Esf-MLS/SchSylv</i> ,
19. <i>SchSylv/Str</i> ,	45. <i>Esf-SchSylv/Sch</i>	71. <i>Esf-MLS/SchComp</i> ,
20. <i>SchSylv/StrUpr</i> ,	46. <i>Esf-SchSylv/SchSylv</i> ,	72. <i>Esf-MLS/SchCompZm</i> ,
21. <i>SchSylv/Sch</i> ,	47. <i>Esf-SchSylv/SchComp</i> ,	73. <i>Esf-SchCompZm/Str</i> ,
22. <i>SchSylv/SchSylv</i> ,	48. <i>Esf-SchSylv/SchCompZm</i> ,	74. <i>Esf-SchCompZm/StrUpr</i> ,
23. <i>SchSylv/SchComp</i> ,	49. <i>Esf-StrUpr/Str</i> ,	75. <i>Esf-SchCompZm/Sch</i>
24. <i>SchSylv/SchCompZm</i> ,	50. <i>Esf-StrUpr/StrUpr</i> ,	76. <i>Esf-SchCompZm/SchSylv</i> ,
25. <i>SchComp/Str</i> ,	51. <i>Esf-StrUpr/Sch</i>	77. <i>Esf-SchCompZm/SchComp</i> ,
26. <i>SchComp/StrUpr</i> ,	52. <i>Esf-StrUpr/SchSylv</i> ,	78. <i>Esf-SchCompZm/SchCompZm</i> ,

RYSUNEK 83. METODY DEKOMPOZYCJI DWUPOZIOMOWYCH



RYSUNEK 84. PRZYKŁADOWE BŁĘDY APROKSYMACJI MODELI ZREDUKOWANYCH DLA RÓŻNYCH METOD DEKOMPOZYCJI DWUPOZIOMEJ

Dekompozycje *Str/Str*, *Str/SchSylv*, *SchComp/Str* oraz *Sch/SchSylv* charakteryzują się większym błędem aproksymacji. Metody dekompozycji wykorzystujące na pierwszym poziomie dekompozycję *slow-fast* dają w zdecydowanej większości przypadków modele stabilne z wyłączeniem dekompozycji *Esf-StrUpr*, *Esf-Str* oraz kombinacji: *Esf-SchComp/SchSylv*, *Esf-Sch/Sch*. Podobna zależność może zostać zaobserwowana w przypadku realizacji dekompozycji *Sch* oraz *SchSylv* na pierwszym poziomie dekompozycji. Poprawne wyniki w odniesieniu do stabilności zostały również uzyskane w przypadku zastosowania dekompozycji *Str/Str*, *Str/Sch* oraz *Str/SchSylv* (około połowa uzyskanych modeli stabilnych). Pozostałe metody dekompozycji dają w większości modele niestabilne, a uzyskanie modeli stabilnych wymaga stosowania zredukowanych modeli podsystemów wysokiego rzędu oraz aproksymacji charakterystyk częstotliwościowych w szerokim zakresie częstotliwości.

TABELA 3. SUMARYCZNE WYNIKI OKREŚLAJĄCE MINIMALNY BŁĄD MODELU ZREDUKOWANEGO ORAZ PROCENT MODELI STABILNYCH DLA ROZWAŻANYCH KOMBINACJI DEKOMPOZYCJI

Poziom 2 \ Poziom 1	<i>Str</i>	<i>StrUpr</i>	<i>Sch</i>	<i>SchSylv</i>	<i>SchComp</i>	<i>SchCompZm</i>
<i>Str</i>	8.5749e-004 46,73%	1.7836 11,44%	0.0952 52,94%	0.0042 57,52% ¹	0.4784 10,46%	1.5266 7,52%
<i>StrUpr</i>	3.0707 21,57%	Zer.	15.9794 38,89% ¹	2.0424 18,95% ¹	Niestab.	Niestab.
<i>Sch</i>	Inter.	1.8630 100%	2.3293 82,35% ¹	0.0079 100%	36.6113 100%	22.2704 100%
<i>SchSylv</i>	Inter.	22.2802 100%	30.0504 86,93% ¹	0.2518 100%	0.3735 100%	1.7241 100%
<i>SchComp</i>	0.0022 12,42%	Niestab.	0.0429 19,28%	0.0224 8,17%	0.0053 2,94%	9.6166 0,98%
<i>SchCompZm</i>	Inter.	Niestab.	3.4963 12,42%	1.1658 14,38%	Niestab.	Niestab.
<i>Esf-SchComp</i>	Inter.	7.5237e-005 97,71%	0.0012 92,48%	4.8033e-006 11,44%	7.3717e-005 99,67%	1.0992e-004 97,39%
<i>Esf-SchSylv</i>	Inter.	4.6082e-005 100%	Niestab.	1.6115e-007 100%	2.0256e-005 100%	1.2767e-004 100%
<i>Esf-StrUpr</i>	Inter.	Niestab.	4.4429 0%	- ¹	8.2573 0%	5.2770 2,61%
<i>Esf-Str</i>	Inter.	Inter.	Inter.	Inter.	Inter.	Inter.
<i>Esf-Sch</i>	Inter.	1.1713 100%	Niestab.	2.5394e-005 100%	37.7615 100%	41.4899 100%
<i>Esf-MLS</i>	Inter.	0.2330 100%	0.2329 100%	0.2327 100%	0.2308 100%	0.2331 95,75%
<i>Esf-SchCompZm</i>	Inter.	0.0872 100%	5.3188 100%	0.0080 97,71%	0.0540 100%	0.0430 100%

W wyniku przeprowadzonych obliczeń stwierdzono również niepraktyczność dekompozycji strukturalnej w odniesieniu do dekompozycji modelu z macierzą *A* zawierającą znaczną liczbę elementów niezerowych znajdujących się w blokach *A*₁₂ i/lub *A*₂₁, które miałyby stanowić interfejsy podsystemów. Przypadek taki wymusza, zgodnie z algorytmem realizacji dekompozycji strukturalnej, uwzględnianie poszczególnych powiązań między

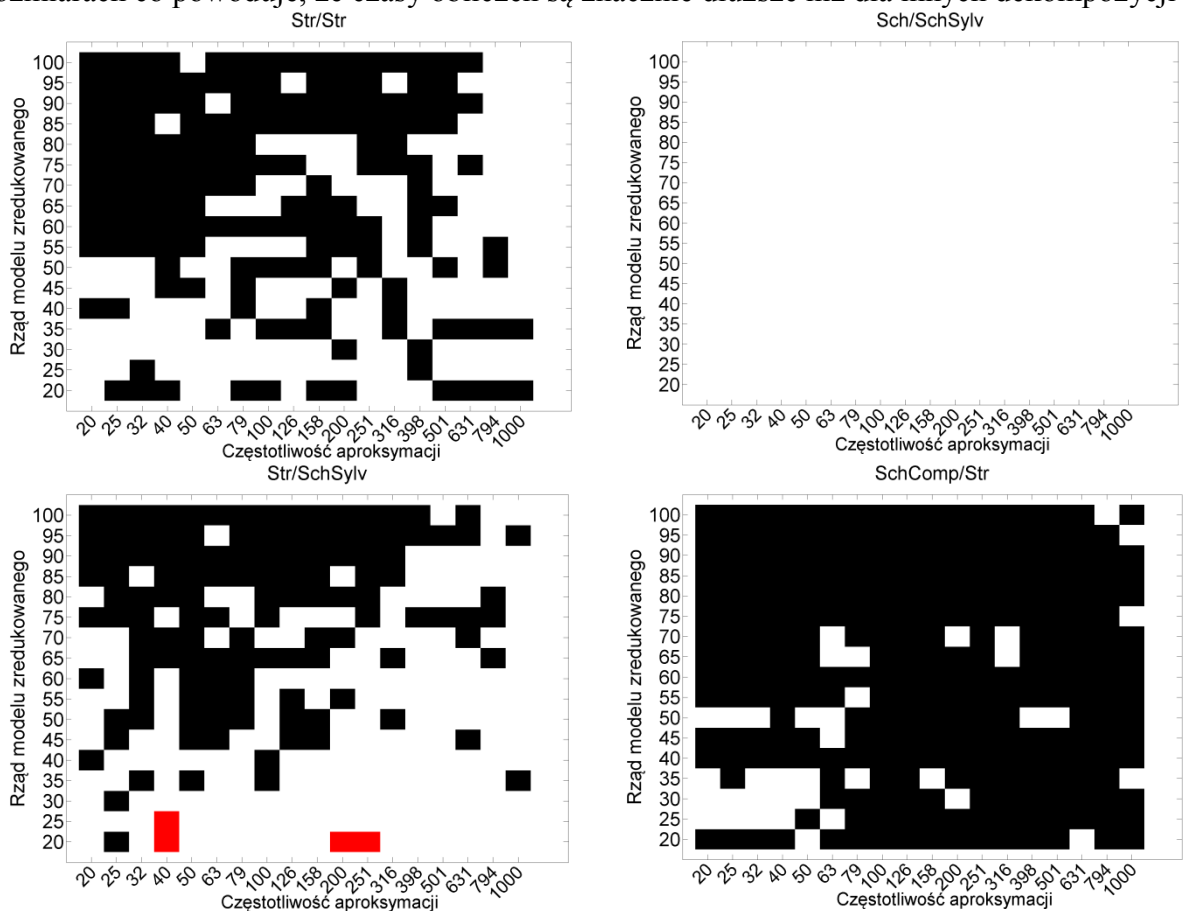
¹ część modeli niezrealizowanych z powodu złego uwarunkowania macierzy modeli zdekomponowanych

podsystemami w macierzach wejściowych i wyjściowych. Prowadzi to do konieczności doboru parametrów wielu filtrów wyjściowych w algorytmie ewolucyjnym przybornika eAMOR i powoduje znaczący wzrost czasu koniecznego do realizacji procesu redukcji. W tabeli 3 kombinacje związane z opisanym problemem oznaczono poprzez Inter.

Kolejny problem, który eliminuje wykorzystanie danej kombinacji dekompozycji jest związany z możliwością uzyskania niestabilnego modelu podsystemu na etapie samej dekompozycji. Związane jest to z realizacją algorytmów dekompozycyjnych, modyfikujących macierze wyznaczanych podsystemów, i koniecznością rozpatrywania uzyskanych modeli podsystemów bez uwzględniania interfejsów je łączących (redukcji podlegają wyłącznie wydzielone podsystemy). Kombinacje dekompozycji, w których stwierdzono wystąpienie opisanego problemu zostały oznaczone w tabeli 3 poprzez Niestab.

Dodatkowo dla kombinacji dekompozycji *StrUpr/StrUpr* dla rozważanego przypadku redukcji modelu strefy dogrzewu zaobserwowano problem związany z uzyskaniem modeli podsystemów z zerowymi macierzami C. Związane jest to ze strukturą macierzy C modelu oryginalnego, która posiada trzy wartości niezerowe w kolumnach numer 2, 586 i 588. Algorytm dekompozycji wydzielił w związku z tym 4 modele, z których dwa środkowe będą posiadały zerową macierz wyjść.

Na rys. 85 przedstawiono mapy stabilności modeli zredukowanych, uzyskane na podstawie wybranych połączeń zdekomponowanych modeli zredukowanych przy zastosowaniu różnych metod dekompozycji. Metody redukcji wykorzystujące dekompozycję slow-fast na pierwszym poziomie dekompozycji dają modele o znacznie różniących się rozmiarach co powoduje, że czasy obliczeń są znacznie dłuższe niż dla innych dekompozycji



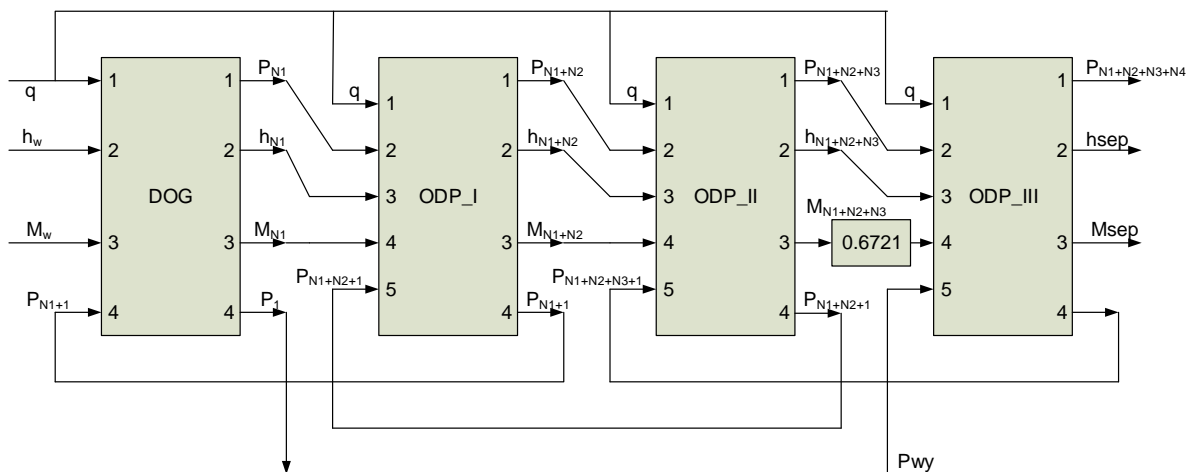
RYSUNEK 85. MAPY STABILNOŚCI MODELI ZREDUKOWANYCH DLA WYBRANYCH DEKOMPOZYCJI DWUPOZIOMOWYCH (kolor biały – model stabilny, kolor czarny – model niestabilny, kolor czerwony – model niezrealizowany z powodu błędów numerycznych)

Podsumowując uzyskane wyniki można stwierdzić, że dla modelu strefy dogrzewu najkorzystniejsze są następujące kombinacje dekompozycji:

- Str/Str – niski błąd aproksymacji, prosty interfejs między podsystemami, w celu zapewnienia stabilności konieczny wysoki zakres aproksymacji charakterystyk częstotliwościowych (ok. 1000 rad/s),
- $Str/SchSylv$ – akceptowalny błąd aproksymacji, w celu zapewnienia stabilności konieczny wysoki zakres aproksymacji (ok. 1000 rad/s), konieczne rozwiązanie równania Sylvester’a (czas realizacji) i dekompozycja Schur’a
- Sch/Sch – akceptowalny błąd aproksymacji, zapewnia stabilność modelu zredukowanego (brak sprzężeń zwrotnych), konieczność rozwiązania równania Sylvester’a i dekompozycji Schur’a,
- $Esf-SchSylv/SchSylv$ – niski błąd aproksymacji, zapewnia stabilność modelu zredukowanego, konieczne dwukrotne rozwiązanie równania Sylvester’a i dekompozycji Schur’a, długi czas realizacji redukcji ze względu na niesymetryczny podział na podsystemy,
- $Esf-SchSylv/SchComp$ – niski błąd aproksymacji, zapewnia stabilność modelu zredukowanego, rozwiązanie równania Sylvester’a i dekompozycja Schur’a, długi czas realizacji redukcji ze względu na niesymetryczny podział na podsystemy,
- $Esf-Sch/SchSylv$ – niski błąd aproksymacji, zapewnia stabilność modelu zredukowanego, rozwiązanie równania Sylvester’a i dekompozycja Schur’a, długi czas realizacji redukcji ze względu na niesymetryczny podział na podsystemy.

7.2. Modele zredukowane podsystemów rur ekranowych

Modele stref: dogrzewu, odparowania I, odparowania II oraz odparowania III połączone są zgodnie ze schematem przedstawionym na rysunku 86.



RYSUNEK 86. SCHEMAT ŁĄCZENIA MODELI PODSYSTEMÓW RUR EKRAWYCH

7.2.1. Wybór metod dekompozycji

Przy wyborze metod dekompozycji dla realizacji redukcji podsystemów rur ekranowych przyjęto następujące założenia. Spośród wszystkich przeanalizowanych dekompozycji dwupoziomowych zostają wybrane trzy typy dekompozycji slow-fast oraz trzy typy z pozostałych metod. Ocena jakości poszczególnych metod została dokonana na podstawie danych zawartych w tabeli 3. Przy wyborze metod slow-fast założono konieczność spełnienia wymagań stabilności na poziomie 100% oraz najmniejszego błędu aproksymacji spośród metod spełniających wcześniejszy warunek. Metody spełniające założone wymagania

to: *Esf-SchSylv/SchSylv*, *Esf-SchSylv/SchComp*, *Esf-Sch/SchSylv*. Przy wyborze metod nie należących do grupy slow-fast założono konieczność spełnienia wymagań stabilności na poziomie minimum 40% oraz błędu aproksymacji nie przekraczającego 1%. Metody spełniające wspomniane wymagania to: *Str/SchSylv*, *Sch/SchSylv*, *Str/Str*.

7.2.2. Parametry redukcji

Badania przeprowadzono dla limitu 800 redukcji algorytmu ewolucyjnego (500 redukcji dla algorytmu wielopunktowego, 300 redukcji dla algorytmu jednopunktowego) oraz limitu 150 redukcji dla algorytmu poszukiwania lokalnego. W przypadku dekompozycji nie należących do grupy *Esf*, założono podział modelu na cztery równe części. Dla dekompozycji z grupy *Esf* wydzielone zostają 3 podsystemy – jeden, stanowiący część szybką - uzyskany w wyniku zmodyfikowanej dekompozycji Epsilon, dwa pozostałe stanowią dalszy podział podsystemu wolnego, z wykorzystaniem dekompozycji realizowanej na drugim poziomie. Ze względu na niesymetryczny podział modelu w wyniku dekompozycji należących do grupy *Esf*, złożoność obliczeniowa procesu redukcji jest bliska wykorzystaniu dekompozycji jednopoziomowej wydzielającej dwa podsystemy.

Ze względu na potrzebę zachowania stabilności zredukowanego modelu rur ekranowych (silne sprzężenia zwrotne między podsystemami), model oryginalny został zredukowany z częstotliwością aproksymacji charakterystyk częstotliwościowych od 0,001 do 1000 rad/s dla dekompozycji *Str/SchSylv*, *Sch/SchSylv*, *Str/Str*. Dla dekompozycji *Esf-SchSylv/SchSylv*, *Esf-SchSylv/SchComp*, *Esf-Sch/SchSylv* część szybka została aproksymowana z zakresem częstotliwości od 0,001 rad/s do 500 rad/s, natomiast część wolna od 0,001 do 100 rad/s. Podczas redukcji *Esf* przyjęto docelowy rząd części szybkiej wynoszący 90. Pozostałą część modelu zredukowanego stanowi wynik redukcji części wolnej.

7.2.3. Modele zredukowane strefy dogrzewu

Oryginalny model strefy dogrzewu o rzędzie równym 3300 stanowi model opisany rzadkimi macierzami stanu. Wielkości wejściowe i wyjściowe strefy dogrzewu przedstawiono na rysunku 86.

W tabeli 4 zostały zebrane wybrane wyniki dla redukcji modelu strefy dogrzewu, każdy przypadek analizowany jest pod względem błędu średniokwadratowego względnego oraz stabilności. Ze względu na zakres adekwatności modeli oryginalnych, miara błędu modeli zredukowanych podawana jest w zakresie częstotliwości od 0,001 do 20 rad/s.

Dekompozycja strukturalna zastosowana na obydwu poziomach dekompozycji daje dokładne modele zredukowane (niski błąd aproksymacji), a błąd aproksymacji maleje wraz z rzędem modelu zredukowanego. Aproksymacja charakterystyk częstotliwościowych podsystemów w zakresie do 1000 rad/s daje pewność uzyskania stabilnego modelu zredukowanego.

Powiązanie dekompozycji strukturalnej (pierwszy poziom) z dekompozycją *SchSylv* pozwala uzyskać zadawalający błąd aproksymacji (dla wyższych rzędów modeli zredukowanych), jednak (szczególnie dla modeli zredukowanych wyższego rzędu) daje niestabilny model zredukowany. Potwierdzenie tego faktu można zauważyć na rys. 85.

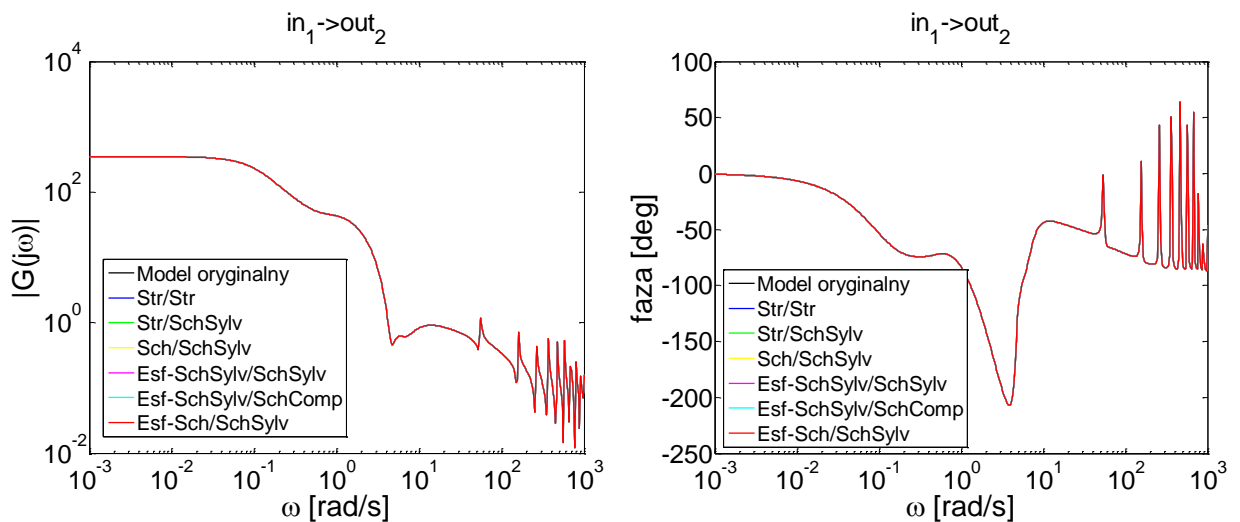
Powiązanie dekompozycji Schura (pierwszy poziom) z dekompozycją *SchSylv* pozwala uzyskać zadawalający błąd aproksymacji (dla wyższych rzędów modeli zredukowanych), przy zapewnieniu stabilności modelu zredukowanego.

TABELA 4. TABELA WYNIKÓW DLA REDUKCJI MODELU STREFY DOGRZEWU

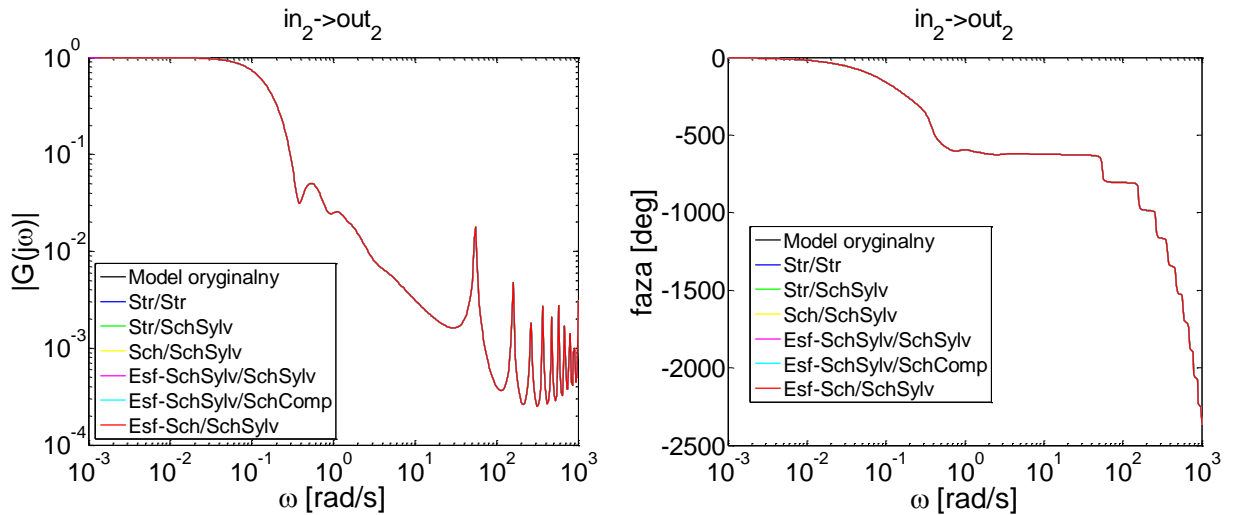
Rząd modelu zredukowanego	Metoda dekompozycji	Str/Str	Str/SchSylv	Sch/SchSylv	Esf-SchSylv/SchSylv	Esf-SchSylv/SchComp	Esf-Sch/SchSylv
		Błąd i stabilność	Błąd i stabilność	Błąd i stabilność	Błąd i stabilność	Błąd i stabilność	Błąd i stabilność
Rząd = 107		3,3762E-02 ✓	18.73563 *	18.30905 ✓	86,53987 ✓	11,91293 ✓	740,0146 ✓
Rząd = 200		1,4200E-05 ✓	0.002247 ✓	0.042893 ✓	0,001306 ✓	0,002151 ✓	0,002513 ✓
Rząd = 220		3,4246E-04 ✓	0.000497 *	0.0101 ✓	0,001306 ✓	0,000264 ✓	0,000331 ✓
Rząd = 242		5,4241E-06 ✓	0.000683 ✓	0.000493 ✓	0,001307 ✓	9,98E-06 ✓	0,000219 ✓
Rząd = 252		2,9064E-07 ✓	1,1453E-04 ✓	8,2824E-03 ✓	0,001305 ✓	8,19E-06 ✓	0,000102 ✓
Rząd = 268		2,3016E-06 ✓	6,5063E-04 ✓	9,4814E-04 ✓	0,001306 ✓	2,21E-05 ✓	1,35E-05 ✓
Rząd = 275		2,0202E-07 ✓	7,0081E-04 ✓	1,7112E-02 ✓	0,001306 ✓	4,93E-06 ✓	2,58E-05 ✓
Rząd = 292		1,5765E-06 ✓	7,4410E-04 *	0.000613 ✓	0,001305 ✓	2,14E-06 ✓	0,000319 ✓
Rząd = 322		8,6495E-08 ✓	6,6579E-04 *	0.000992 ✓	0,001306 ✓	2,18E-06 ✓	0,006146 ✓
Rząd = 356		2,7590E-09 ✓	5,3442E-05 *	0.000985 ✓	0,001293 ✓	4,21E-08 ✓	0,002935 ✓

Spośród metod dekompozycji opartych na dekompozycji slow-fast (pierwszy poziom) najlepsze wyniki uzyskuje się z zastosowaniem metody *Esf-SchSylv/SchComp*. Jednak wszystkie metody dają modele stabilne z akceptowalnym błędem. Podniesienie rzędu zredukowanej części szybkiej z 90 do 110 daje poprawę rezultatów dla dekompozycji *Esf-Sch/SchSylv* – błąd średniokwadratowy względny wynosi 6.6294e-004. W przypadku pozostałych metod podniesienie rzędu zredukowanej części szybkiej nie wpływa na poprawę rezultatów.

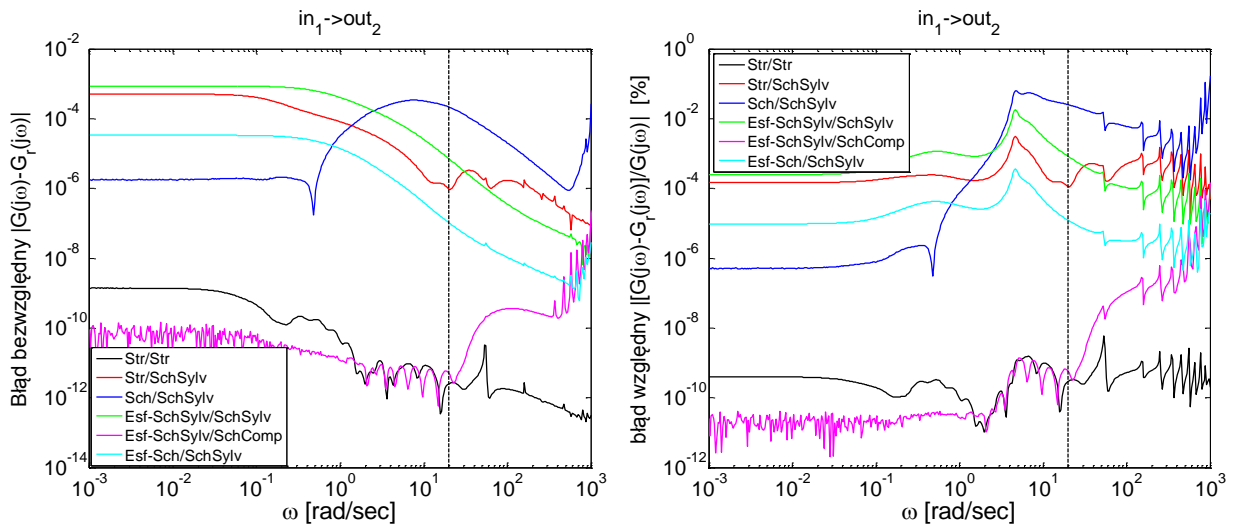
Na rys. 87- 88 przedstawiono charakterystyki częstotliwościowe uzyskane na podstawie modelu pierwotnego strefy dogrzewu oraz modeli zredukowanych, oznaczonych kolorem zielonym w tabeli 4, dla wybranych wielkości wejściowych i wyjściowych. Na rys. 89 - 90 przedstawiono wykresy błędu względnego i bezwzględnego dla torów $\Delta q \rightarrow \Delta h_{NI}$ oraz $\Delta h_w \rightarrow \Delta h_{NI}$. Pionowa linia przerywana wskazuje zakres adekwatności modelu.



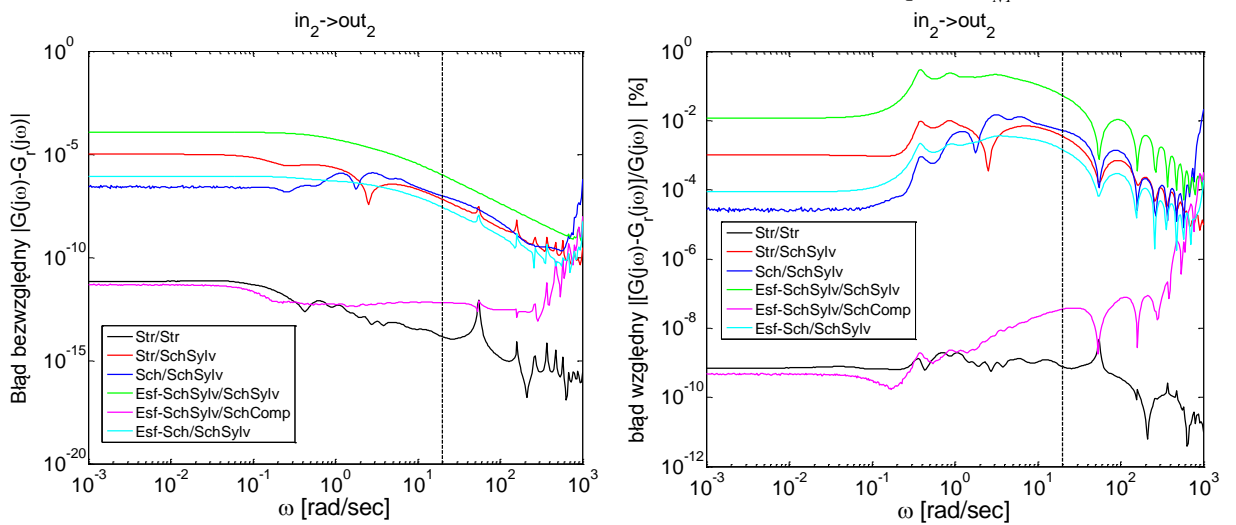
RYSUNEK 87. PRZYKŁADOWE WYKRESY CHARAKTERYSTYK CZĘSTOTLIWOŚCIOWYCH DLA MODELU ORYGINALNEGO I MODELI ZREDUKOWANYCH STREFY DOGRZEWU (tor: $\Delta q \rightarrow \Delta h_{NI}$)



RYSUNEK 88. PRZYKŁADOWE WYKRESY CHARAKTERYSTYK CZĘSTOTLIWOŚCIOWYCH DLA MODELU ORYGINALNEGO I MODELI ZREDUKOWANYCH STREFY DOGRZEWU (tor: $\Delta h_w \rightarrow \Delta h_{NI}$)



RYSUNEK 89. PRZYKŁADOWE WYKRESY BEZWZGLĘDNEGO I WZGLĘDNEGO BŁĘDU APROKSYMACJI DLA MODELI STREFY DOGRZEWU (tor: $\Delta q \sim \rightarrow \Delta h_{NI}$)



RYSUNEK 90. PRZYKŁADOWE WYKRESY BEZWZGLĘDNEGO I WZGLĘDNEGO BŁĘDU APROKSYMACJI DLA MODELI STREFY DOGRZEWU (tor: $\Delta h_w \rightarrow \Delta h_{NI}$)

7.2.4. Modele zredukowane strefy odparowania I

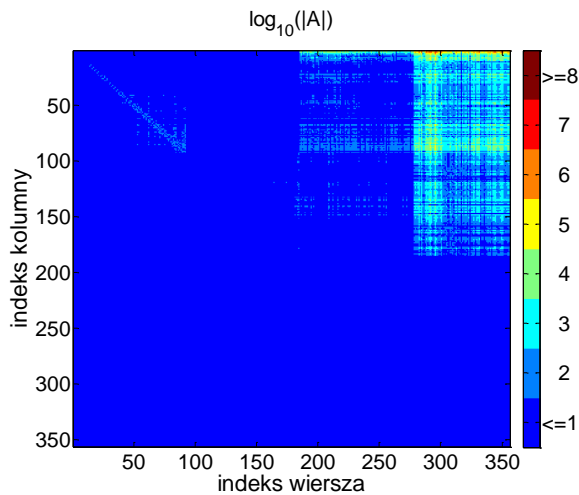
Podobnie jak w przypadku strefy dogrzewu, strefa odparowania I jest opisana macierzami stanu będącymi macierzami rzadkimi. Strefa odparowania I stanowi model o 3300 zmiennych stanu. W tabeli 5 zostały zebrane wybrane wyniki dla redukcji modelu strefy dogrzewu, każdy przypadek analizowany jest pod względem błędu średniokwadratowego względnego oraz stabilności. Ze względu na zakres adekwatności modeli oryginalnych, miara błędu modeli zredukowanych podawana jest w zakresie częstotliwości od 0,001 do 20 rad/s.

TABELA 5. TABELA WYNIKÓW DLA REDUKCJI MODELU STREFY ODPAROWANIA I

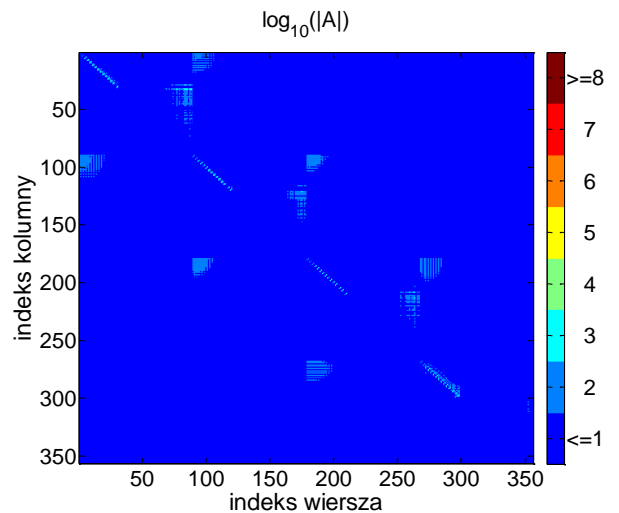
Rząd modelu zredu.	Metoda dekompozycji	Str/Str	Str/SchSylv	Sch/SchSylv	Esf-SchSylv/SchSylv	Esf-SchSylv/SchComp	Esf-Sch/SchSylv
		Błąd i stabilność	Błąd i stabilność	Błąd i stabilność	Błąd i stabilność	Błąd i stabilność	Błąd i stabilność
Rząd = 130		5,4309E-03 ✓	241,9053 ✓	8,6833E+01 ✓	1,701476 ✓	0,593072 ✓	3,923934 ✓
Rząd = 164		2,5906E-05 ✓	1548,949 ✓	7,4708E+01 ✓	0,00119 ✓	0,006219 ✓	0,810001 ✓
Rząd = 172		1,6618E-05 ✓	274,4947 ✓	7,4632E+01 ✓	4,74E-05 ✓	0,008555 ✓	0,805895 ✓
Rząd = 180		7,9449E-06 ✓	62,15605 *	7,4711E+01 ✓	0,000209 ✓	0,001594 ✓	0,806368 ✓
Rząd = 218		1,3924E-07 *	148,5734 ✓	7,4721E+01 ✓	3,23E-05 ✓	0,000231 ✓	0,806883 ✓
Rząd = 242		4,4113E-06 ✓	8,38011 ✓	7,4691E+01 ✓	3,23E-05 ✓	3,24E-05 ✓	0,806853 ✓
Rząd = 304		3,5172E-08 ✓	0,141075 ✓	3,4532E+01 ✓	3,23E-05 ✓	3,23E-05 ✓	0,806853 ✓
Rząd = 322		2,7889E-09 ✓	0,019717 *	40,73508 ✓	3,23E-05 ✓	3,23E-05 ✓	0,80685 ✓
Rząd = 356		1,2517E-10 ✓	7,4324e-004 ✓	4,336455 ✓	3,23E-05 ✓	3,23E-05 ✓	0,806856 ✓
Rząd = 368		7,6918E-11 ✓	3,2387E-05 *	5,595542 ✓	3,23E-05 ✓	3,23E-05 ✓	0,806861 ✓

Dekompozycja strukturalna zastosowana na obydwu poziomach dekompozycji daje dokładne modele zredukowane, a błąd aproksymacji maleje przy wzroście rzędu modelu zredukowanego. Aproksymacja charakterystyk częstotliwościowych podsystemów w zakresie do $\omega_{max} = 1000 \text{ rad/s}$ zapewnią uzyskanie stabilnego modelu zredukowanego.

Metody Str/SchSylv oraz Sch/SchSylv dają modele zredukowane charakteryzujące się wysokim błędem aproksymacji, a także niestabilnością (metoda Str/SchSylv). Wysoki błąd aproksymacji dla metody Sch/SchSylv wynika z właściwości interfejsu. Przy niewielkich błędach aproksymacji dla zdekomponowanych modeli zredukowanych, interfejsy dla metody Sch/SchSylv charakteryzują się bardzo silnym wzmocnieniem powodując, że scalony model zredukowany charakteryzuje się bardzo dużym błędem. Na rys. 91 przedstawiono strukturę macierzy A pod względem wartości elementów. Dla porównania, rysunek 92 został sporządzony dla modeli zredukowanych w wyniku dekompozycji Str/Str.



RYSUNEK 91. STRUKTURA MACIERZY A PO REDUKCJI Z ZASTOSOWANIEM DEKOMPOZYCJI *Sch/SchSylv*

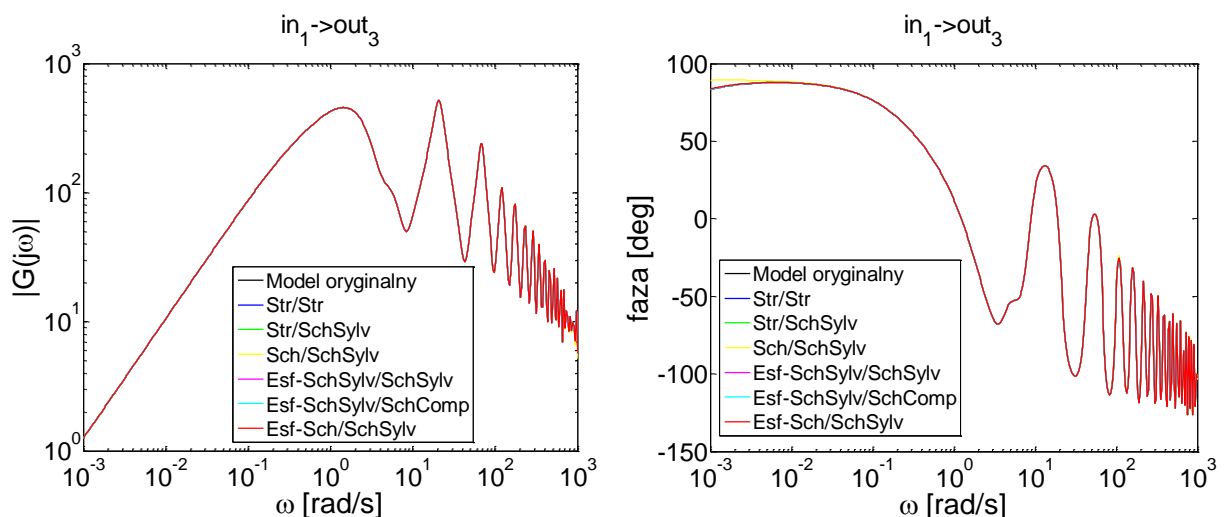


RYSUNEK 92. STRUKTURA MACIERZY A PO REDUKCJI Z ZASTOSOWANIEM DEKOMPOZYCJI *Str/Str*

Metody oparte na dekompozycji slow-fast (*Esf-SchSylv/SchSylv* oraz *Esf-SchSylv/SchComp*) charakteryzują się niskim błędem aproksymacji oraz stabilnością modeli zredukowanych (niezależnie od rzędu modelu zredukowanego). Metoda *Esf-Sch/SchSylv* daje wysoki błąd aproksymacji. Podniesienie rzędu części szybkiej z 90 do 110 znacząco wpływa na poprawę błędu aproksymacji dla wszystkich metod *Esf*. Błąd średniokwadratowy względny wynosi wówczas:

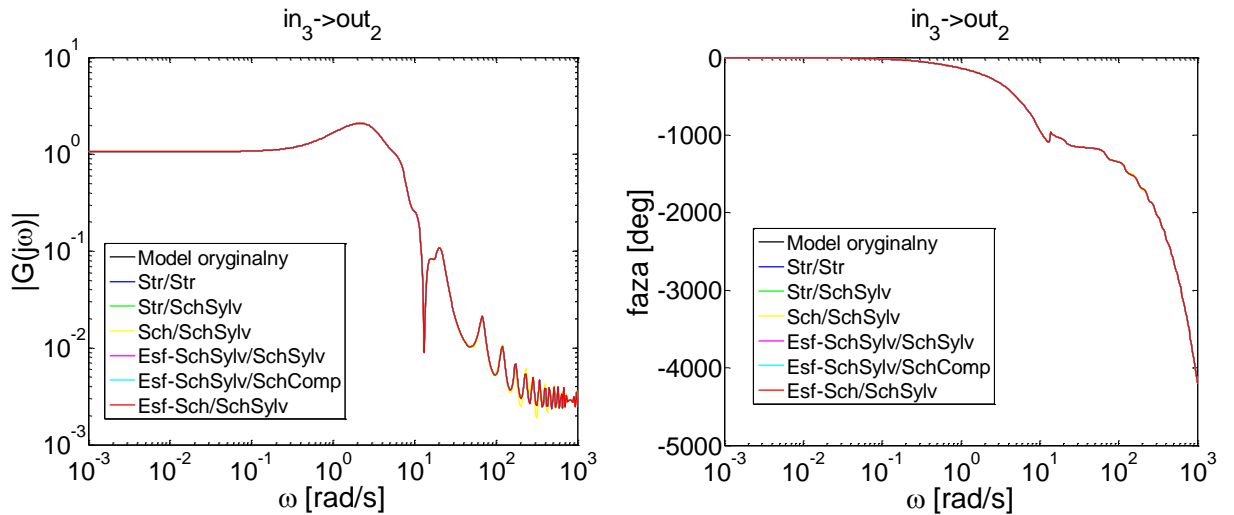
- 1.7605e-007 dla *Esf-SchSylv/SchSylv*,
- 4.4928e-006 dla *Esf-SchSylv/SchComp*,
- 7.1214e-006 dla *Esf-Sch/SchSylv*.

Na rys. 93 - 94 przedstawiono charakterystyki częstotliwościowe uzyskane na podstawie modelu pierwotnego strefy odparowania I oraz modeli zredukowanych, oznaczonych zielonym kolorem w tabeli 5, dla wybranych wielkości wejściowych i wyjściowych



RYSUNEK 93. PRZYKŁADOWE WYKRESY CHARAKTERYSTYK CZĘSTOTLIWOŚCIOWYCH DLA MODELU ORYGINALNEGO I MODELÓW ZREDUKOWANYCH STREFY ODPAROWANIA I (tor:

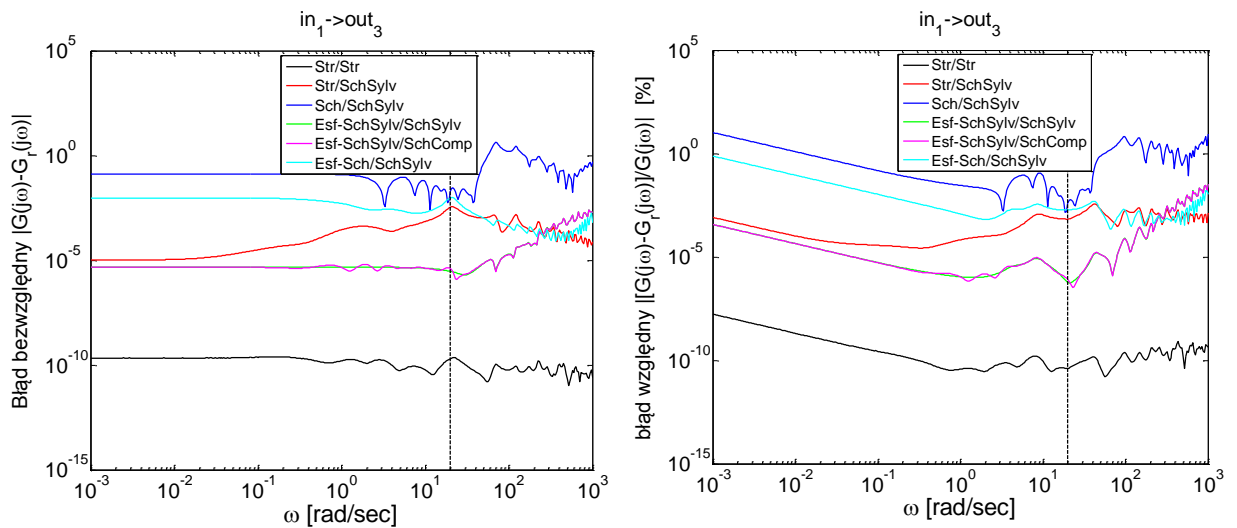
$$\Delta q^{\sim} \rightarrow \Delta M_{II})$$



RYSUNEK 94. PRZYKŁADOWE WYKRESY CHARAKTERYSTYK CZĘSTOTLIWOŚCIOWYCH DLA MODELU ORYGINALNEGO I MODELI ZREDUKOWANYCH STREFY ODPAROWANIA I (tor:

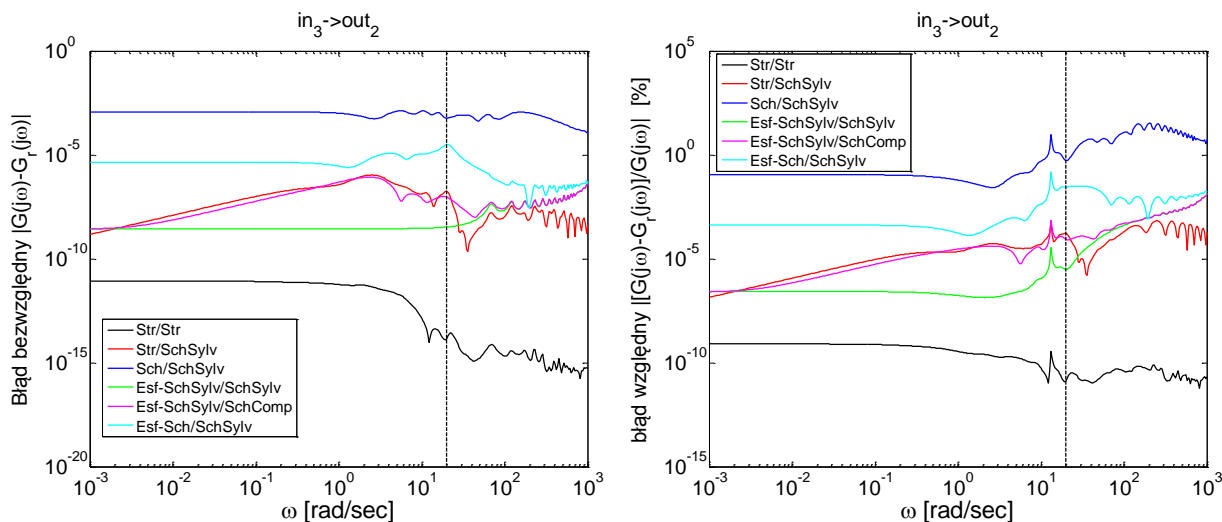
$$\Delta h_{NI} \rightarrow \Delta h_{II})$$

Na rys. 95 - 96 przedstawiono błąd aproksymacji charakterystyk częstotliwościowych modeli zredukowanych, oznaczonych zielonym kolorem w tabeli 5, dla wybranych wielkości wejściowych i wyjściowych.



RYSUNEK 95. PRZYKŁADOWE WYKRESY BEZWZGLĘDNEGO I WZGLĘDNEGO BŁĘDU

$$\text{APROKSYMACJI DLA MODELI STREFY ODPAROWANIA I (tor: } \Delta q^{\sim} \rightarrow \Delta M_{II})$$



RYSUNEK 96. PRZYKŁADOWE WYKRESY BEZWZGLĘDNEGO I WZGLĘDNEGO BŁĘDU APROKSYMACJI DLA MODELI STREFY ODPAROWANIA I (tor: $\Delta h_{NI} \rightarrow \Delta h_{II}$)

7.2.5. Modele zredukowane strefy odparowania II

Model strefy odparowania II posiada 3300 zmiennych stanu. Podobnie jak dwie poprzednie strefy opisany jest macierzami stanu o postaci rzadkiej. W tabeli 6 zostały zebrane wybrane wyniki dla redukcji modelu strefy odparowania II, każdy przypadek analizowany jest pod względem błędu średniokwadratowego względnego oraz stabilności. Ze względu na zakres adekwatności modeli oryginalnych, miara błędu modeli zredukowanych podawana jest w zakresie częstotliwości od 0,001 do 20 rad/s.

TABELA 6. TABELA WYNIKÓW DLA REDUKCJI STREFY ODPAROWANIA II

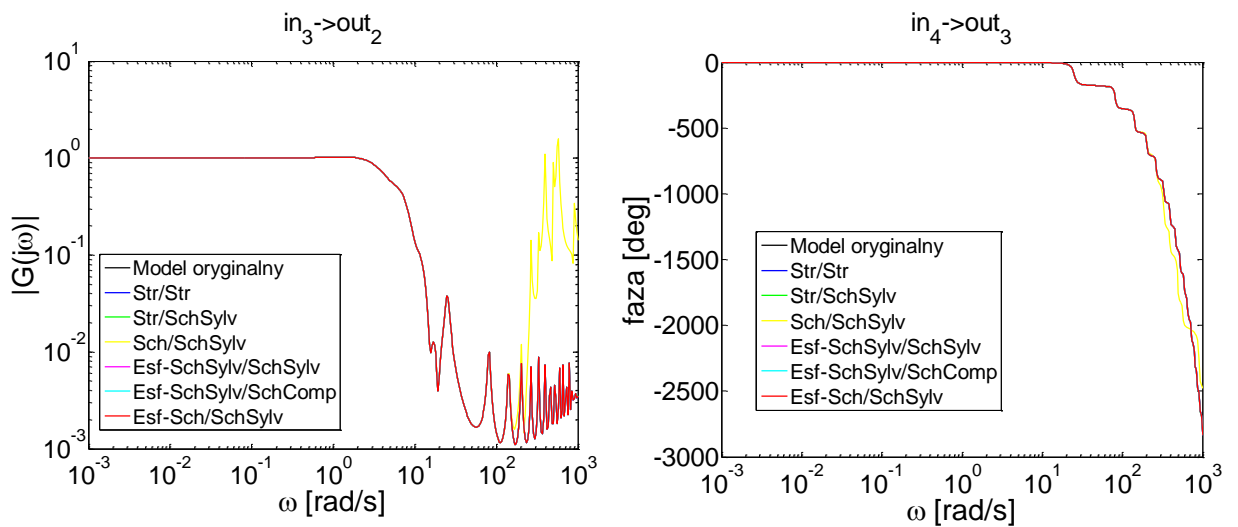
Rząd modelu zredu.	Metoda dekompozycji	Str/Str	Str/SchSylv	Sch/SchSylv	Esf-SchSylv/SchSylv	Esf-SchSylv/SchComp	Esf-Sch/SchSylv
		Błąd i stabilność	Błąd i stabilność	Błąd i stabilność	Błąd i stabilność	Błąd i stabilność	Błąd i stabilność
Rząd = 120		1,9442E-01 ✓	283,0714 ✓	2,0989E+02 ✓	1,042104 ✓	0,740643 ✓	3,346483 ✓
Rząd = 145		2,8158E-03 ✓	291,0076 ✓	2,9062E-01 ✓	0,008404 ✓	0,008481 ✓	0,031283 ✓
Rząd = 164		7,6444E-05 ✓	127,5486 ✓	4,4900E-01 ✓	0,008101 ✓	0,008131 ✓	0,00414 ✓
Rząd = 198		1,8977E-07 ✓	36,72919 ✓	4,4212E-03 ✓	0,008101 ✓	0,008101 ✓	0,003758 ✓
Rząd = 264		7,8670E-06 ✓	0,785729 ✓	1,2183E-02 ✓	0,008101 ✓	0,008101 ✓	0,003759 ✓
Rząd = 275		1,4506E-06 ✓	0,119602 ✓	5,1509E-02 ✓	0,008101 ✓	0,008101 ✓	0,003763 ✓
Rząd = 322		1,6717E-06 ✓	3,3678E-02 ✓	4,2497E-02 ✓	0,008101 ✓	0,008101 ✓	0,003757 ✓
Rząd = 332		1,1589E-07 ✓	9,6346E-04 *	2,9619E-02 ✓	0,008101 ✓	0,008101 ✓	0,003757 ✓
Rząd = 356		3,3049E-10 ✓	1,5772E-06 *	2,3732E+01 ✓	0,008101 ✓	0,008101 ✓	0,003758 ✓
Rząd = 404		2,0413E-10 *	5,5573E-07 ✓	5,2174E+02 ✓	0,008101 ✓	0,008101 *	0,003757 ✓

Na podstawie tabeli 6 można stwierdzić, że najlepsze wyniki redukcji otrzymuje się w rezultacie zastosowania dekompozycji strukturalnej na obydwu poziomach. W wyniku wzrostu rzędu modelu zredukowanego maleje względny błąd średniokwadratowy

aproxymacji i osiąga poziom 10^{-10} . Metody dekompozycji *Sch* oraz *SchSylv* dają duży błąd aproxymacji. Wszystkie metody dekompozycji slow-fast pozwalają na sprowadzenie błędu aproxymacji do określonego poziomu ($8 \cdot 10^{-3}$, $3 \cdot 10^{-3}$), bez możliwości zmniejszenia tego błędu mimo zwiększania rzędu części wolnej modelu zredukowanego. Po zwiększeniu rzędu części szybkiej modelu zredukowanego z 90 do 110, błąd aproxymacji gwałtownie maleje dla wszystkich metod. Wynosi on wówczas:

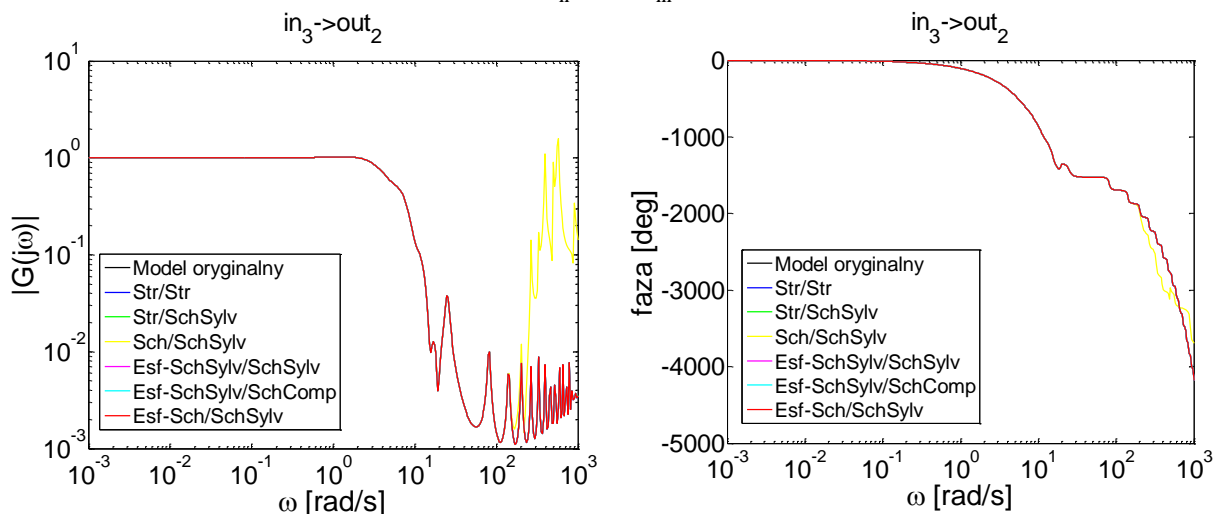
- $2.2945e-009$ dla *Esf-SchSylv/SchSylv*,
- $1.2975e-009$ dla *Esf-SchSylv/SchComp*,
- $7.9200e-005$ dla *Esf-Sch/SchSylv*.

Na rys. 97 - 98 przedstawiono charakterystyki częstotliwościowe uzyskane na podstawie modelu pierwotnego strefy odparowania II oraz modeli zredukowanych, oznaczonych zielonym kolorem w tabeli 6, dla wybranych wielkości wejściowych i wyjściowych.



RYSUNEK 97. PRZYKŁADOWE WYKRESY CHARAKTERYSTYK CZĘSTOTLIWOŚCIOWYCH DLA MODELU ORYGINALNEGO I MODELI ZREDUKOWANYCH STREFY ODPAROWANIA II (tor:

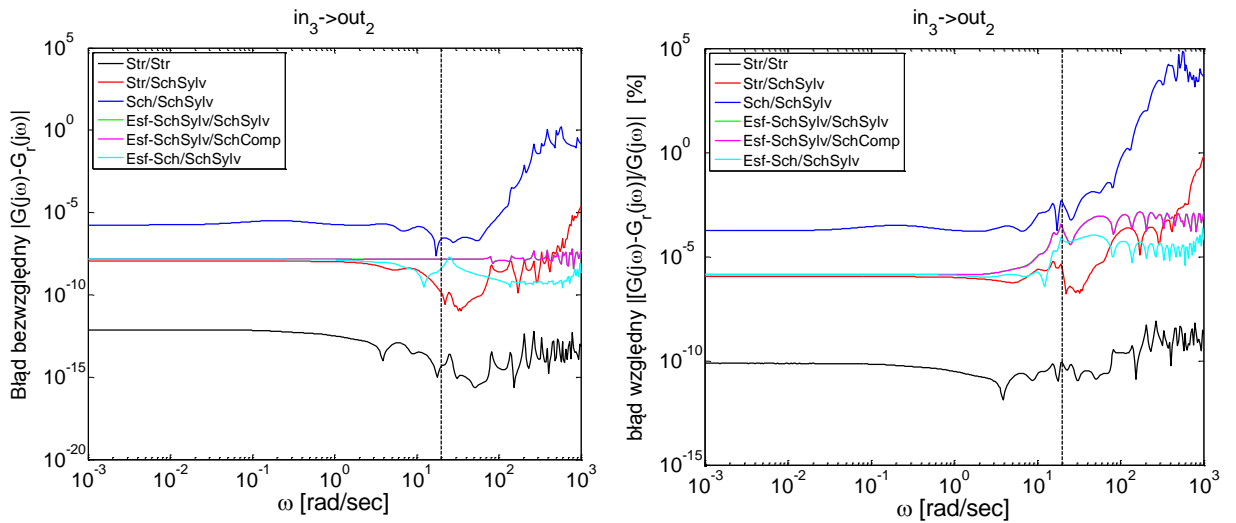
$$\Delta h_{II} \rightarrow \Delta h_{III})$$



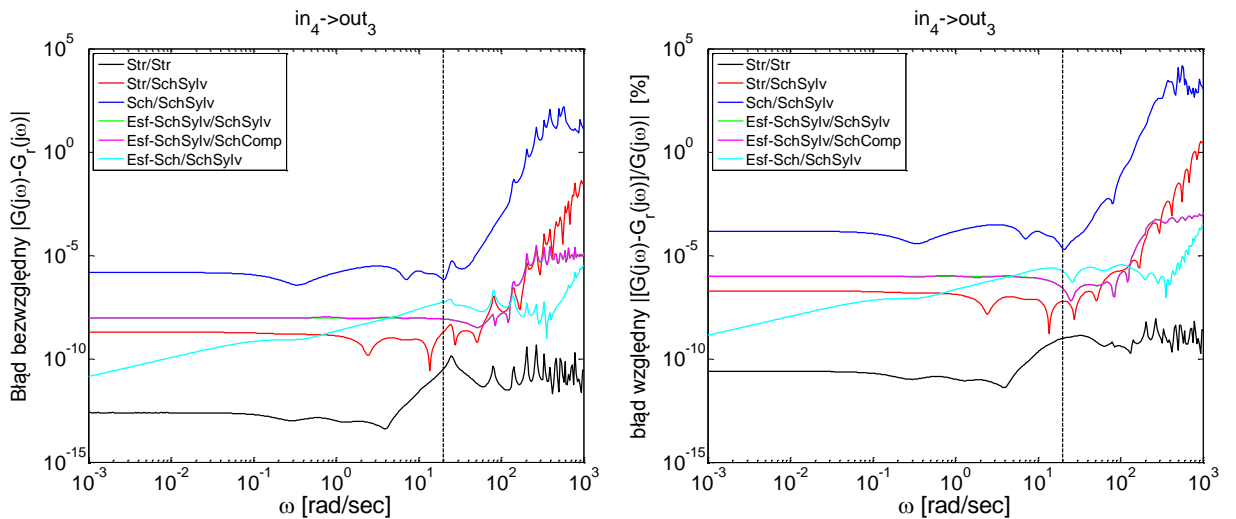
RYSUNEK 98. PRZYKŁADOWE WYKRESY CHARAKTERYSTYK CZĘSTOTLIWOŚCIOWYCH DLA MODELU ORYGINALNEGO I MODELI ZREDUKOWANYCH STREFY ODPAROWANIA II (tor:

$$\Delta M_{we} \rightarrow \Delta M_{wy})$$

Na rys. 99 - 100 przedstawiono błąd aproksymacji charakterystyk częstotliwościowych modeli zredukowanych, oznaczonych zielonym kolorem w tabeli 6, dla wybranych wielkości wejściowych i wyjściowych.



RYСУNEK 99. PRZYKŁADOWE WYKRESY BEZWZGLĘDNEGO I WZGLĘDNEGO BŁĘDU APROKSYMACJI DLA MODELI STREFY ODPAROWANIA II (tor: $\Delta h_{II} \rightarrow \Delta h_{III}$)



RYСУNEK 100. PRZYKŁADOWE WYKRESY BEZWZGLĘDNEGO I WZGLĘDNEGO BŁĘDU APROKSYMACJI DLA MODELI STREFY ODPAROWANIA II (tor: $\Delta M_{II} \rightarrow \Delta M_{III}$)

7.2.6. Modele zredukowane strefy odparowania III

Model strefy odparowania III posiada 3300 zmiennych stanu, opisany jest macierzami stanu o postaci rzadkiej. W tabeli 7 zostały zebrane wybrane wyniki dla redukcji modelu strefy odparowania III, każdy przypadek analizowany jest pod względem błędu średniokwadratowego względnego oraz stabilności. Ze względu na zakres adekwatności modeli oryginalnych, miara błędów modeli zredukowanych podawana jest w zakresie częstotliwości od 0,001 do 20 rad/s.

TABELA 7. TABELA WYNIKÓW DLA REDUKCJI STREFY ODPAROWANIA III

Rząd modelu zredukowanego	Metoda dekompozycji	Str/Str	Str/SchSylv	Sch/SchSylv	Esf-SchSylv/SchSylv	Esf-SchSylv/SchComp	Esf-Sch/SchSylv
		Błąd i stabilność	Błąd i stabilność	Błąd i stabilność	Błąd i stabilność	Błąd i stabilność	Błąd i stabilność
Rząd = 130		3,0375E-02 ✓	97,20862 ✓	9,0627E+01 ✓	0,071699 ✓	0,081922 ✓	0,081 ✓
Rząd = 164		1,8469E-04 ✓	31,66218 ✓	4,4614E-03 ✓	0,000131 ✓	0,000684 ✓	0,002238 ✓
Rząd = 172		3,9281E-01 ✓	23,66265 ✓	8,5803E+01 ✓	0,000134 ✓	0,000538 ✓	0,002228 ✓
Rząd = 242		5,8686E-05 ✓	0,504996 ✓	3,0398E-03 ✓	0,000132 ✓	0,000133 ✓	0,002219 ✓
Rząd = 256		6,4373E-07 ✓	0,49591 *	6,5981E-05 ✓	0,000133 ✓	0,000132 ✓	0,002218 ✓
Rząd = 292		1,1437E-07 ✓	2,5106E-03 *	6,5993E+00 ✓	0,000132 ✓	0,000134 ✓	0,002218 ✓
Rząd = 304		6,5620E-08 ✓	6,9681E-05 ✓	1,2630E-04 ✓	0,000132 ✓	0,000154 ✓	0,002219 ✓
Rząd = 332		7,2637E-09 ✓	1,1427E-04 ✓	3,6455E-03 ✓	0,000132 ✓	0,000133 ✓	0,002219 ✓
Rząd = 356		6,5811E-10 ✓	4,2270E-05 *	7,6341E-03 ✓	0,000132 ✓	0,000134 ✓	0,002218 ✓
Rząd = 368		4,7723E-10 ✓	3,0009E-04 *	5,2004E+01 ✓	0,000132 ✓	0,000132 ✓	0,002218 ✓

Na podstawie tabeli 7 można stwierdzić, że najlepsze wyniki redukcji otrzymuje się w rezultacie zastosowania dekompozycji strukturalnej na obydwu poziomach. W wyniku wzrostu rzędu modeli zredukowanych maleje średniokwadratowy względny błąd aproksymacji i osiąga poziom 10^{-10} .

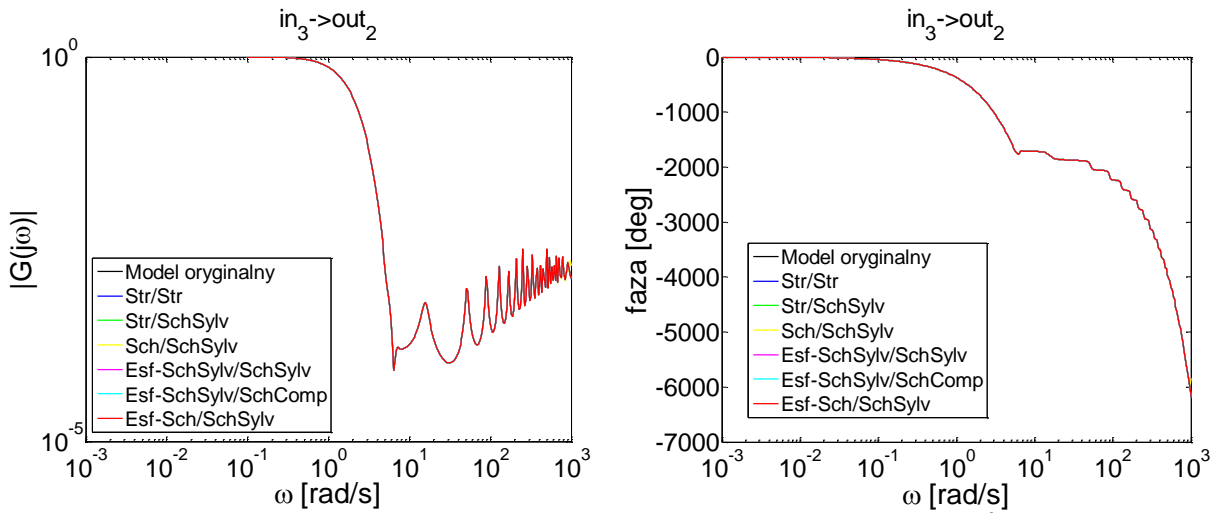
Analogicznie jak dla strefy odparowania I i II, metody dekompozycji *Sch* oraz *SchSylv* dają dla wybranych rzędów modelu zredukowanego akceptowalne błędy aproksymacji.

Wszystkie metody dekompozycji oparte na dekompozycji slow-fast pozwalają na sprowadzenie błędu aproksymacji do niskiego poziomu ($\sim 10^{-4}$). Jednak błąd pozostaje niezmienny przy wzroście rzędu części wolnej modelu zredukowanego. Po zwiększeniu rozmiaru części szybkiej modelu zredukowanego z 90 do 110 analogicznie jak w przypadku strefy odparowania II zauważono znaczny spadek błędu aproksymacji. Wynosi on wówczas:

- 1.5767e-006 dla *Esf-SchSylv/SchSylv*,
- 6.1457e-008 dla *Esf-SchSylv/SchComp*,
- 4.2882e-006 dla *Esf-Sch/SchSylv*.

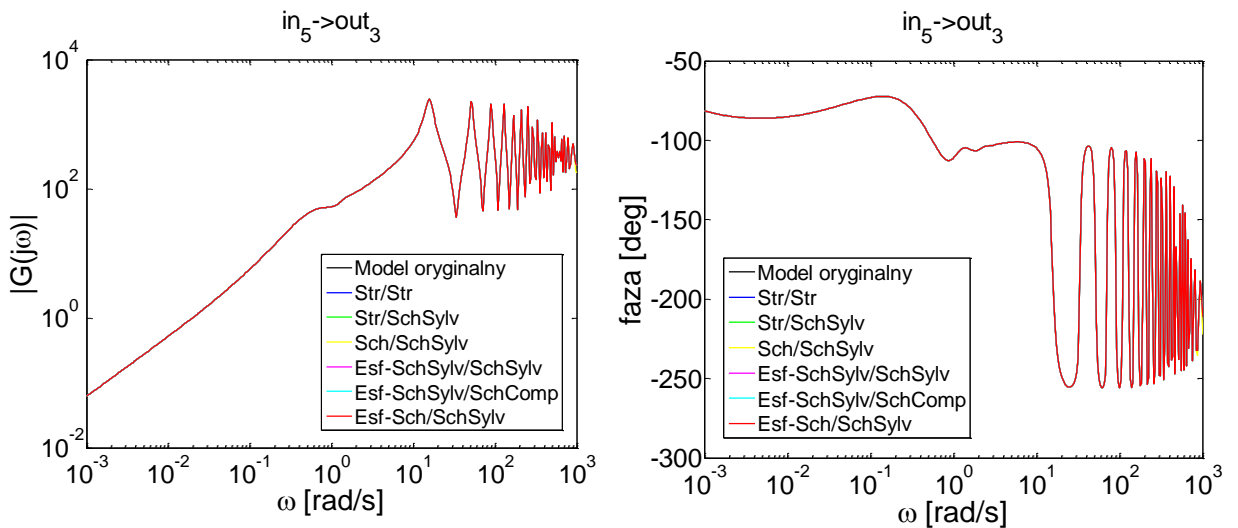
Na rys. 101 - 102 przedstawiono charakterystyki częstotliwościowe uzyskane na podstawie modelu pierwotnego strefy odparowania III oraz modeli zredukowanych, oznaczonych zielonym kolorem w tabeli 7, dla wybranych wielkości wejściowych i wyjściowych.

Na rys. 103 - 104 przedstawiono błąd aproksymacji charakterystyk częstotliwościowych modeli zredukowanych, oznaczonych zielonym kolorem w tabeli 7, dla wybranych wielkości wejściowych i wyjściowych.



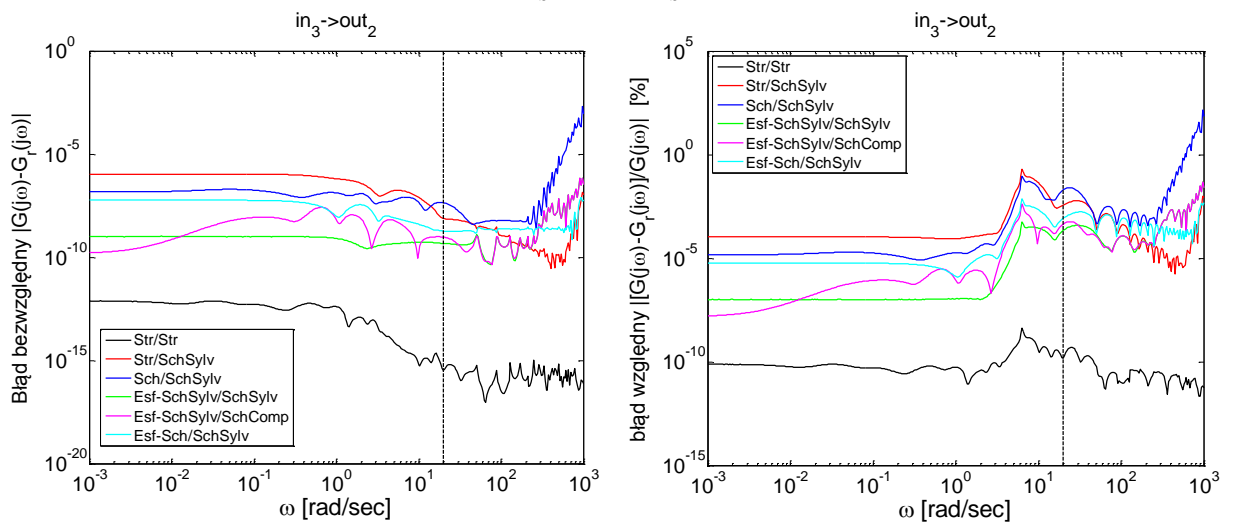
RYСУNEK 101. PRZYKŁADOWE WYKRESY CHARAKTERYSTYK CZĘSTOTLIWOŚCIOWYCH DLA MODELU ORYGINALNEGO I MODELI ZREDUKOWANYCH STREFY ODPAROWANIA III (tor:

$$\Delta h_{III} \rightarrow \Delta h_{SEP}$$



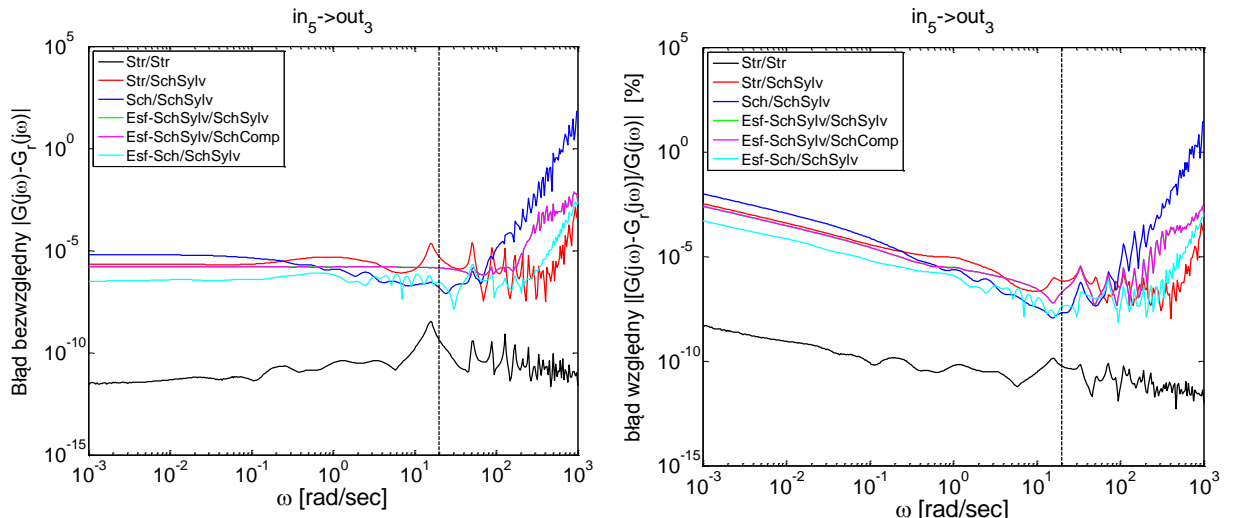
RYСУNEK 102. PRZYKŁADOWE WYKRESY CHARAKTERYSTYK CZĘSTOTLIWOŚCIOWYCH DLA MODELU ORYGINALNEGO I MODELI ZREDUKOWANYCH STREFY ODPAROWANIA III (tor:

$$\Delta P_{SEP} \rightarrow \Delta M_{SEP}$$



RYСУNEK 103. PRZYKŁADOWE WYKRESY BEZWZGLĘDNEGO I WZGLĘDNEGO BŁĘDU

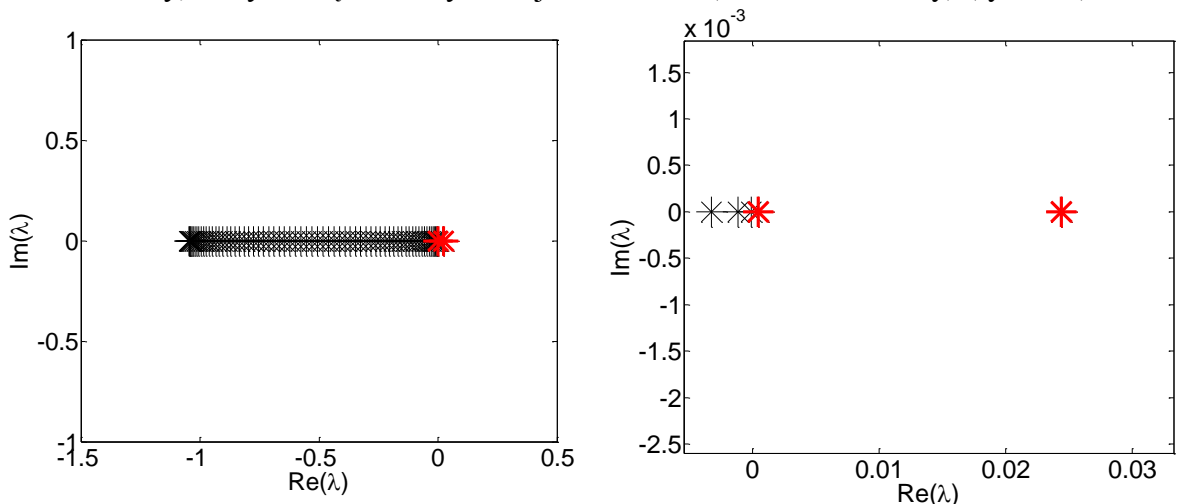
APROKSYMACJI DLA MODELI STREFY ODPAROWANIA III (tor: $\Delta h_{III} \rightarrow \Delta h_{SEP}$)



RYSUNEK 104. PRZYKŁADOWE WYKRESY BEZWZGLĘDNEGO I WZGLĘDNEGO BŁĘDU APROKSYMACJI DLA MODELII STREFY ODPAROWANIA III (tor: $\Delta P_{SEP} \rightarrow \Delta M_{SEP}$)

7.3. Modele zredukowane separatora

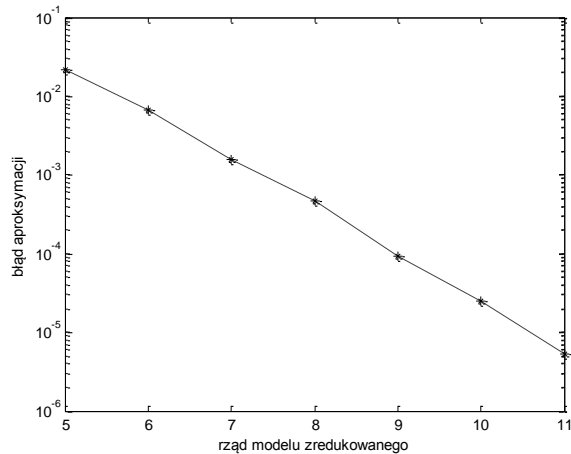
Zbiór wartości własnych macierzy stanu A oryginalnego modelu separatora zawiera dwa elementy, których część rzeczywista jest dodatnia (model niestabilny) (rys. 105).



RYSUNEK 105. WYKRES WARTOŚCI WŁASNYCH MACIERZY STANU A MODELII SEPARATORA

Przed realizacją redukcji modelu separatora, model oryginalny zdekomponowany został na model stabilny (70 zmiennych stanu) oraz model niestabilny (2 zmienne stanu). Operację realizowano z wykorzystaniem polecenia `stabsep` środowiska MATLAB. Operacji redukcji podlega wyłącznie część stabilna.

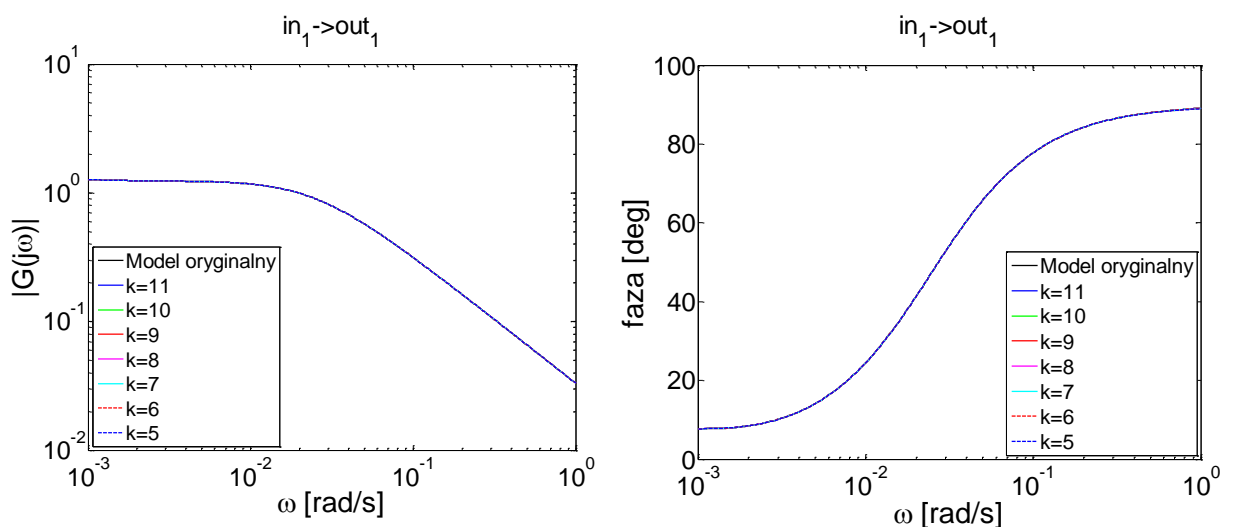
Część stabilną zredukowano z wykorzystaniem przybownika eAMOR w zakresie częstotliwości od 10^{-3} rad/s do 20 rad/s, uzyskując modele o rzędzie od 5 do 11. Wykres względnego błędu średniokwadratowego modelu zredukowanego, w stosunku do modelu oryginalnego (z częścią niestabilną), w zależności od rzędu modelu został zamieszczony na rysunku 106.



RYSUNEK 106. WYKRES BŁĘDU REDUKCJI W ZALEŻNOŚCI OD RZĘDU MODELU ZREDUKOWANEGO DLA MODELU SEPARATORA

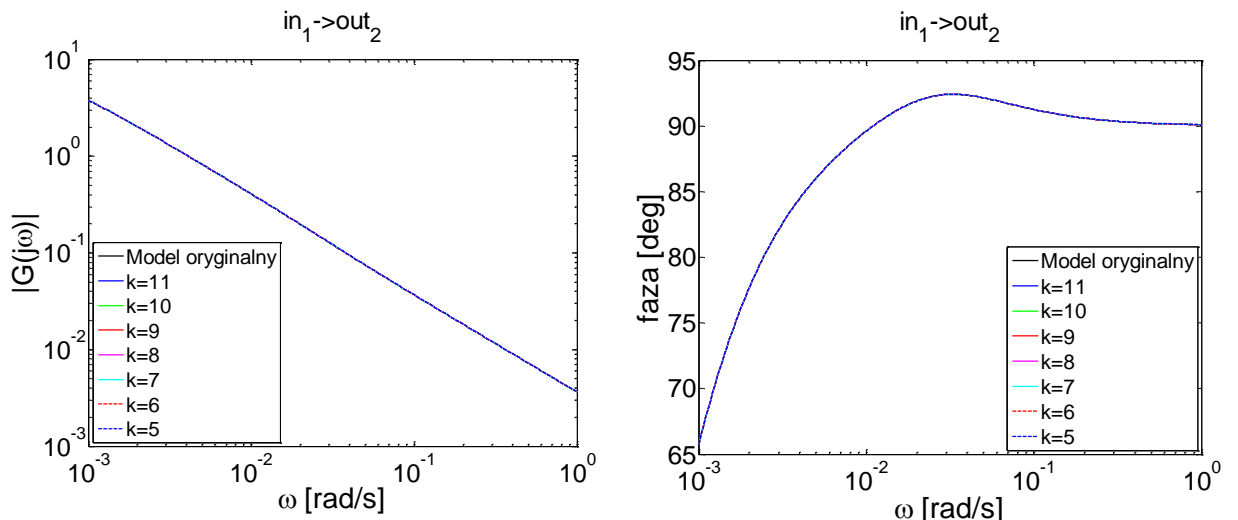
Model matematyczny separatora charakteryzuje się dwoma wielkościami wyjściowymi – ciśnieniem ΔP_{sep} oraz poziomem wody ΔH_{sep} w separatorze. Można powiedzieć, że obydwie wielkości charakteryzują aktualny stan bilansu masy (ΔH_{sep}) oraz bilansu energii (ΔP_{sep}) w parowniku. Obydwie wielkości są bezpośrednio mierzone i wykorzystywane w układzie sterowania. Zerowy biegun modelu separatora (w rzeczywistości niewielka wartość dodatnia) wynika z całkujących właściwości modelu separatora dla wielkości wyjściowej ΔH_{sep} (0,025). Natomiast dodatni biegun modelu separatora wynika z dodatniego sprzężenia zwrotnego, związanego z zależnością parametrów wody i pary od ciśnienia (wzrost ciśnienia powoduje spadek entalpii pary nasyconej suchej $h''(P)$) [149].

Na rys. 107 - 108 porównano charakterystyki częstotliwościowe separatora dla wielkości wyjściowych ΔP_{sep} i ΔH_{sep} oraz wielkości wejściowych ΔM_p oraz ΔM_{kond} , na podstawie modelu oryginalnego oraz modeli zredukowanych.



RYSUNEK 107. PRZYKŁADOWE WYKRESY CHARAKTERYSTYK CZĘSTOTLIWOŚCIOWYCH DLA MODELU ORYGINALNEGO I MODELU ZREDUKOWANYCH SEPARATORA DLA WIELKOŚCI

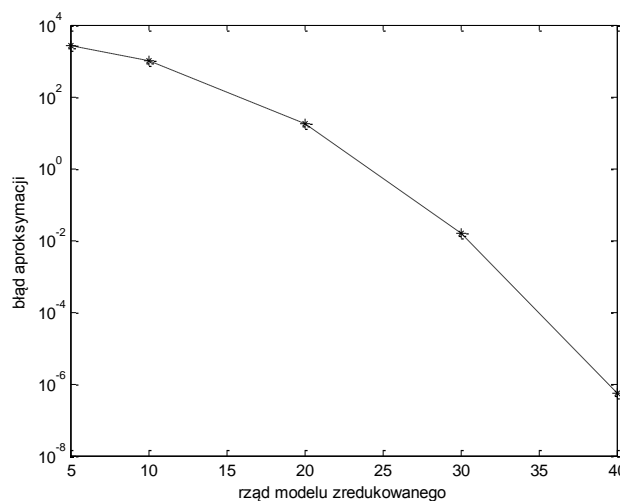
WEJŚCIOWEJ ΔM_p (tor: $\Delta M_p \rightarrow \Delta P_{sep}$)



RYSUNEK 108. PRZYKŁADOWE WYKRESY CHARAKTERYSTYK CZĘSTOTLIWOŚCIOWYCH DLA MODELU ORYGINALNEGO I MODELI ZREDUKOWANYCH SEPARATORA DLA WIELKOŚCI WEJŚCIOWEJ ΔM_p (tor: $\Delta M_p \rightarrow \Delta H_{sep}$)

7.4. Modele zredukowane układu cyrkulacji

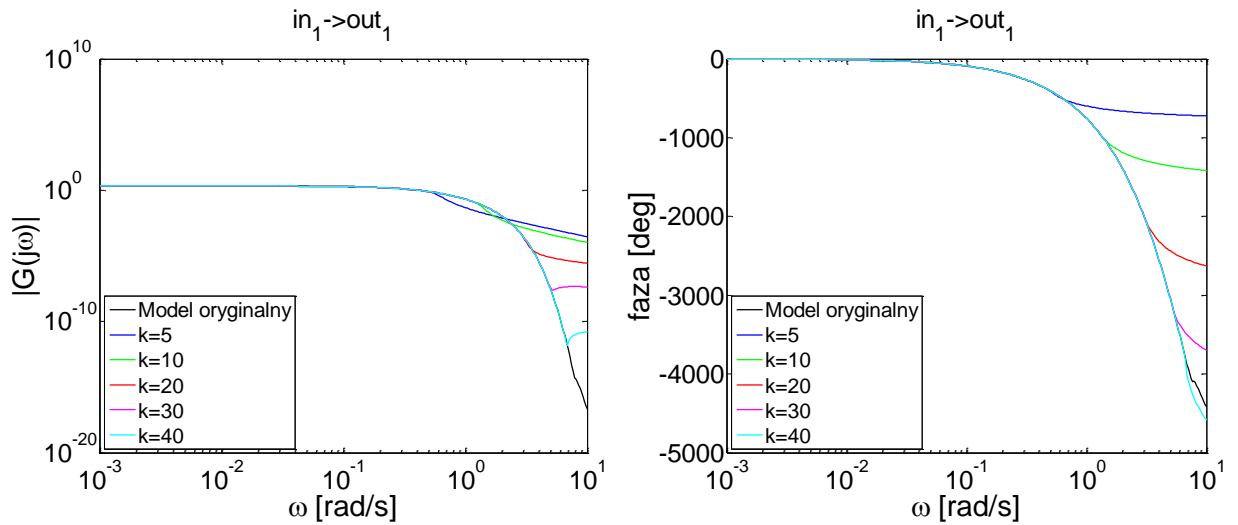
Model układu cyrkulacji, o rozmiarze 612 zmiennych stanu, zredukowano z wykorzystaniem przybornika eAMOR w zakresie częstotliwości od 10^{-3} rad/s do 5 rad/s, uzyskując modele o rozmiarach: 5, 10, 20, 30 i 40. Wykres względnego błędu średniokwadratowego modelu zredukowanego, w stosunku do modelu oryginalnego, w zależności od rzędu modelu został zamieszczony na rysunku 109.



RYSUNEK 109. WYKRES BŁĘDU REDUKCJI W ZALEŻNOŚCI OD RZĘDU MODELU ZREDUKOWANEGO DLA MODELU UKŁADU CYRKULACJI

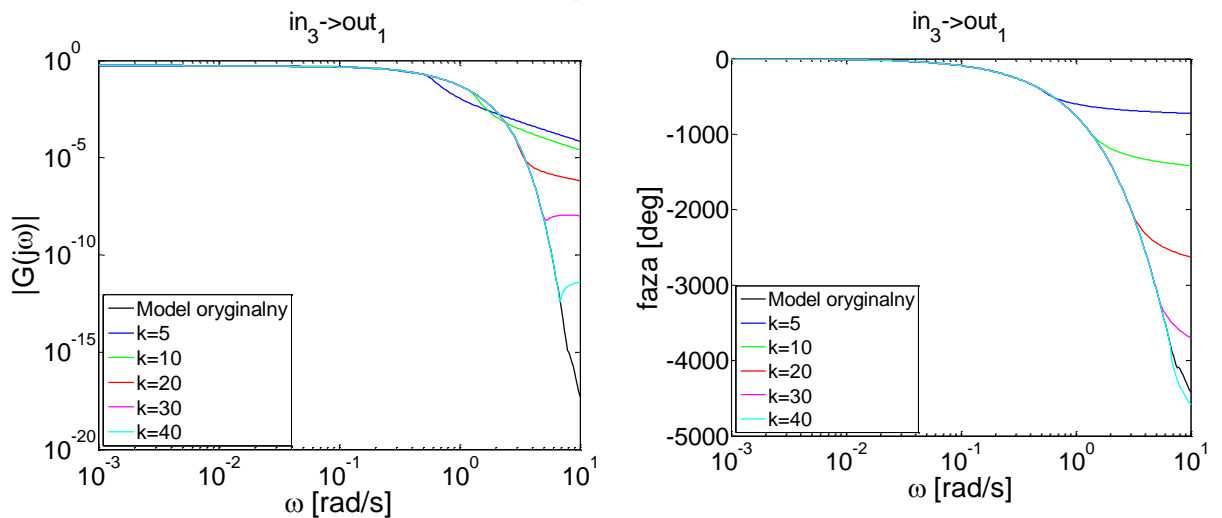
W skład układu cyrkulacji w parowniku wchodzi: filtr, mieszalnik, rurociągi łączące o łącznej długości kilkudziesięciu metrów. Ze względu na dużą pojemność wodną tego układu, oraz masę elementów stalowych, układ cyrkulacji wody w parowniku charakteryzuje się wysoką inercją. Ze względu na dużą wartość zastępczej stałej czasowej układu cyrkulacji wody, moduł charakterystyk częstotliwościowych dla $\omega > 5 \text{ rad/s}$ jest praktycznie zerowy. Z tego powodu ograniczono częstotliwościowy zakres aproksymacji charakterystyk częstotliwościowych dla tego przypadku do 5 rad/s .

Na rys. 110 - 111 przedstawiono charakterystyki częstotliwościowe układu cyrkulacji wody w parowniku dla wybranych wielkości wejściowych i wielkości wyjściowej Δh_w , otrzymane na podstawie modelu oryginalnego oraz modeli zredukowanych.



RYSUNEK 110. PRZYKŁADOWE WYKRESY CHARAKTERYSTYK CZĘSTOTLIWOŚCIOWYCH DLA MODELU ORYGINALNEGO I MODELI ZREDUKOWANYCH UKŁADU CYRKULACJI (tor:

$$\Delta P_{sep} \rightarrow \Delta h_w)$$



RYSUNEK 111. PRZYKŁADOWE WYKRESY CHARAKTERYSTYK CZĘSTOTLIWOŚCIOWYCH DLA MODELU ORYGINALNEGO I MODELI ZREDUKOWANYCH UKŁADU CYRKULACJI (tor:

$$\Delta h_{w\text{eko}} \rightarrow \Delta h_w)$$

7.5. Powtórnie zredukowane modele rur ekranowych

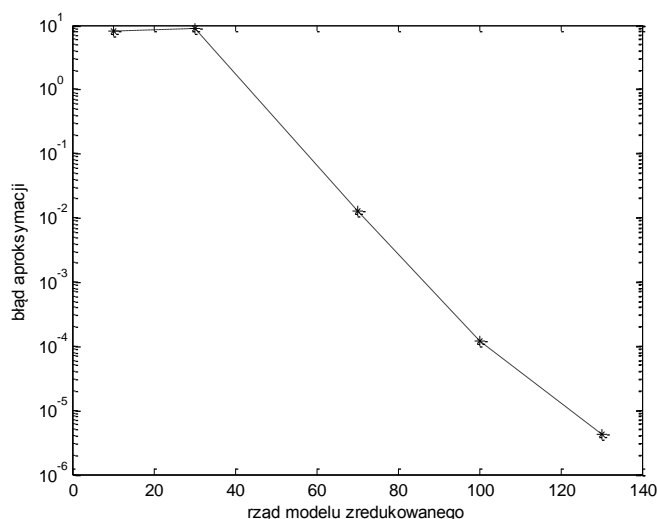
W tabeli 8 zebrano podstawowe informacje na temat wybranych modeli wykorzystanych w celu uzyskania wstępnie zredukowanego modelu rur ekranowych:

TABELA 8. PODSTAWOWE INFORMACJE NA TEMAT WYBRANYCH MODELI WYKORZYSTANYCH W CELU UZYSKANIA WSTĘPNIE ZREDUKOWANEGO MODELU RURU EKRANOWYCH

Model	rzęd	błąd średniokwadratowy	metoda dekompozycji
strefa dogrzewu	356	2,76E-09	<i>Str/Str</i>
strefa odparowania I	368	7,69E-11	<i>Str/Str</i>
strefa odparowania II	356	3,30E-10	<i>Str/Str</i>
strefa odparowania III	368	4,77E-10	<i>Str/Str</i>

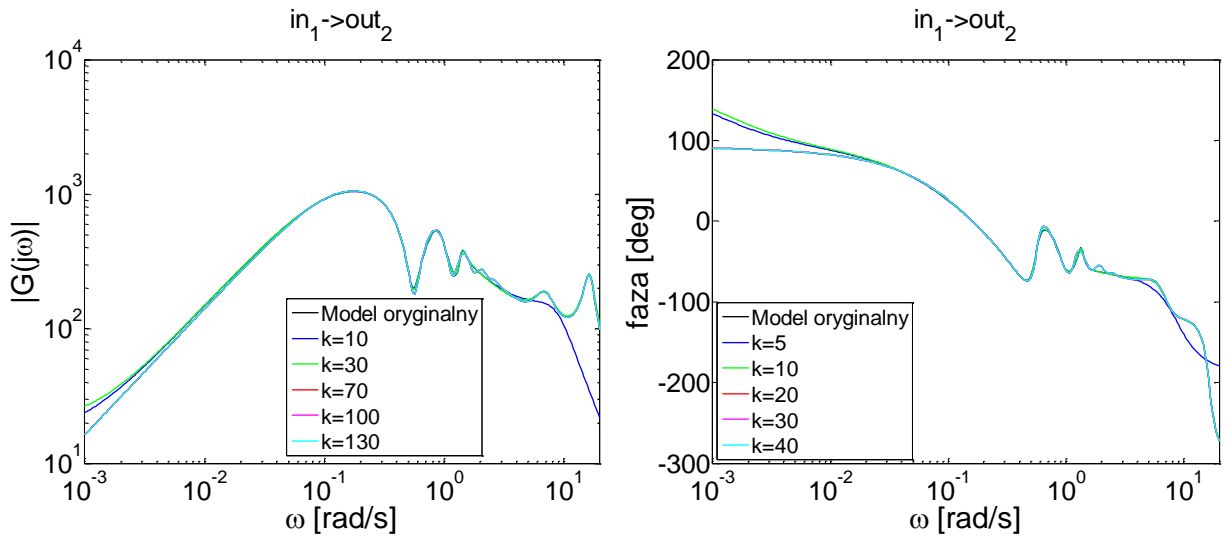
Model rur ekranowych o tak wysokim rozmiarze wynika przede wszystkim z konieczności zapewnienia stabilności modeli zredukowanych, uzyskanych na podstawie scalenia zdekomponowanych modeli zredukowanych podsystemów (zdekomponowane modele zredukowane aproksymują charakterystyki częstotliwościowe podsystemów w zakresie do 10^3 rad/s).

Po scaleniu zredukowanych modeli podsystemów rur ekranowych uzyskuje się model, którego rząd jest zbyt wysoki do dalszych analiz, a szczególnie do projektowania układu sterowania. Z tego powodu uzyskany model rur ekranowych poddano ponownej operacji redukcji z wykorzystaniem przybornika eAMOR w zakresie częstotliwości 10^{-3} -20 rad/s, uzyskując modele rzędu: 10, 30, 70, 100 oraz 130. Średniokwadratowy błąd względny aproksymacji charakterystyk częstotliwościowych modelu powtórnie zredukowanego, w stosunku do modelu pierwotnie zredukowanego, w funkcji rzędu modelu powtórnie zredukowanego przedstawiono na rys. 112.



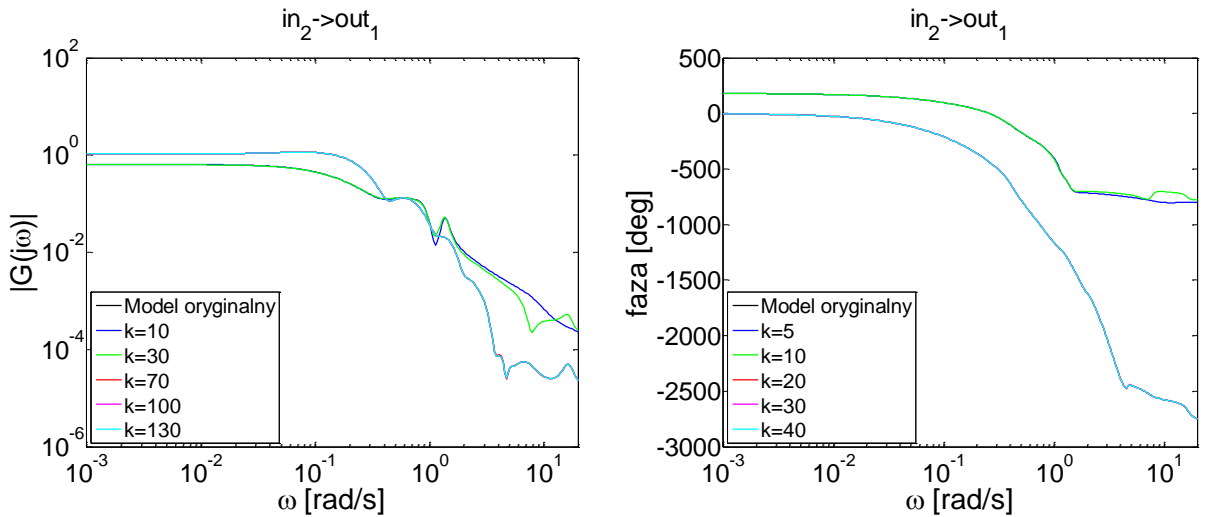
RYSUNEK 112. WYKRESY BŁĘDU REDUKCJI POWTÓRNICZIE ZREDUKOWANEGO MODELU RUR EKRANOWYCH, W STOSUNKU DO MODELU PIERWOTNICZIE ZREDUKOWANEGO W ZALEŻNOŚCI OD RZĘDU MODELU POWTÓRNICZIE ZREDUKOWANEGO

Na rys. 113 - 114 porównano charakterystyki częstotliwościowe modelu rur ekranowych, otrzymane na podstawie modelu pierwotnie zredukowanego oraz modeli powtórnie zredukowanych różnego rzędu, dla wybranych wielkości wejściowych i wyjściowych.



RYСУNEK 113. PRZYKŁADOWE WYKRESY CHARAKTERYSTYK CZĘSTOTLIWOŚCIOWYCH DLA MODELU WSTĘPNIE ZREDUKOWANEGO I MODELI POWTÓRNIE ZREDUKOWANYCH RUR

EKRANOWYCH (tor: $\Delta q \rightarrow \Delta M_{SEP}$)



RYСУNEK 114. PRZYKŁADOWE WYKRESY CHARAKTERYSTYK CZĘSTOTLIWOŚCIOWYCH DLA MODELU WSTĘPNIE ZREDUKOWANEGO I MODELI POWTÓRNIE ZREDUKOWANYCH RUR

EKRANOWYCH (tor: $\Delta h_w \rightarrow \Delta h_{SEP}$)

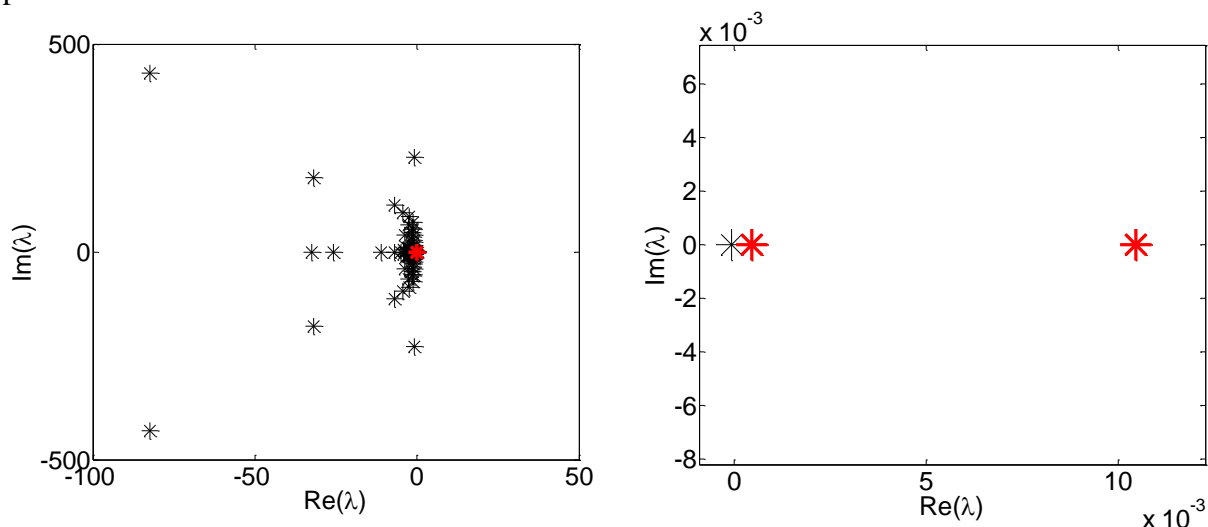
7.6. Modele zredukowane parownika

Do wyznaczenia wstępnie zredukowanego modelu parownika posłużyły zredukowane modele rur ekranowych, separatora oraz układu cyrkulacji. Podstawowe informacje o modelach użytych w celu wyznaczenia wstępnie zredukowanego modelu parownika zostały zamieszczone w tabeli 9.

TABELA 9. PODSTAWOWE INFORMACJE NA TEMAT WYBRANYCH MODELI WYKORZYSTANYCH W CELU UZYSKANIA WSTĘPNIE ZREDUKOWANEGO MODELU PAROWNIKA

model	rzęd	błąd średniokwadratowy
rury ekranowe	130	4.1660e-006
separator	13	5.2915e-006
układ cyrkulacji	40	5.5154e-007

Po połączeniu wymienionych modeli, wstępnie zredukowany model parownika jest niestabilny (dwie wartości własne dodatnie związane z modelem separatora). Na rysunku 115 przedstawiono wykres wartości własnych macierzy stanu A wstępnie zredukowanego modelu parownika.

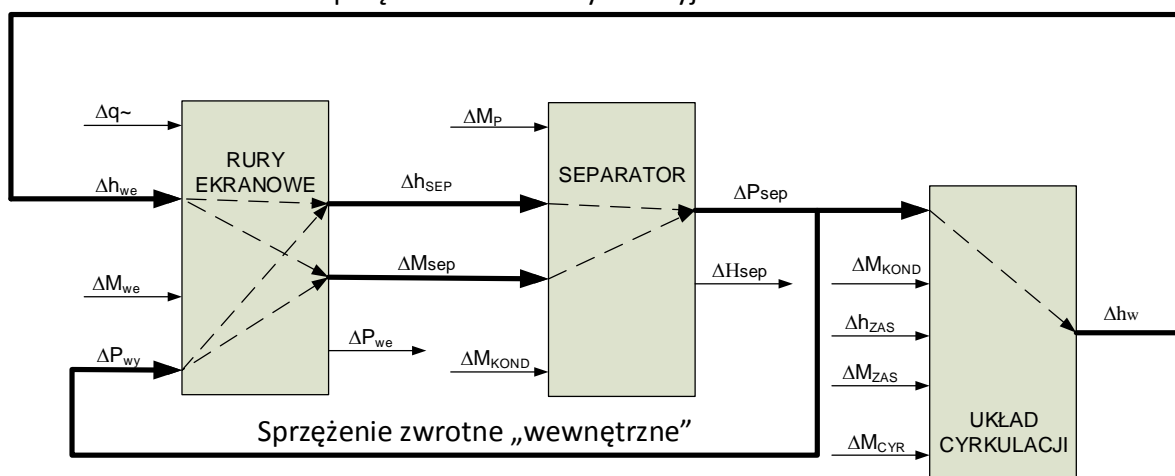


RYSUNEK 115. WYKRES WARTOŚCI WŁASNYCH MACIERZY STANU A WSTĘPNIE ZREDUKOWANEGO MODELU PAROWNIKA

W wyniku operacji scalania, uzyskano model parownika kotła BP-1150 183 rzędu. Model posiada dwa dodatnie bieguny – jeden o wartości około 10^{-4} (praktycznie 0) i drugi o wartości około 10^{-2} . Biegun o wartości około +0,01 jest przesuniętym biegunem separatora o wartości około +0,025 (rys. 105). Przesunięcie bieguna wynika z dwu sprzężeń zwrotnych funkcjonujących w parowniku:

- „wewnętrzne” sprzężenie zwrotne (ujemne) – wynika z oddziaływania ciśnienia w separatorze ΔP_{sep} na przepływ mieszaniny parowo-wodnej do separatora ΔM_{sep} (rys. 116),
- Cyrkulacyjne sprzężenie zwrotne (dodatnie), wynikające z funkcjonowania cyrkulacji w parowniku (rys. 116).

Sprężenie zwrotne cyrkulacyjne



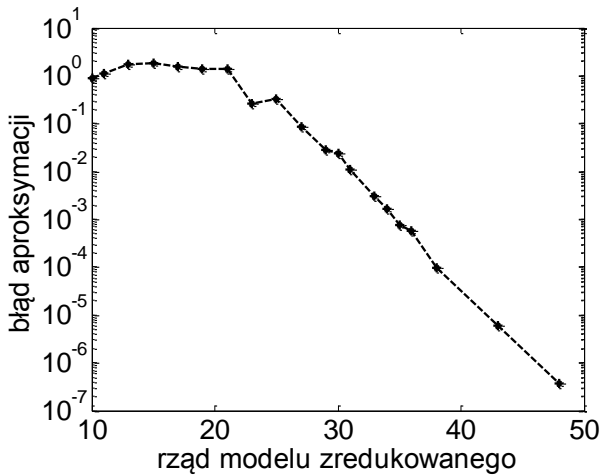
RYSUNEK 116. SPRĘŻENIA ZWROTNE FUNKCJONUJĄCE W PAROWNIKU

Przed realizacją redukcji modelu parownika wykonano dekompozycję modelu na część stabilną i niestabilną. Część stabilna (rozmiar 181 zmiennych stanu) została poddana redukcji

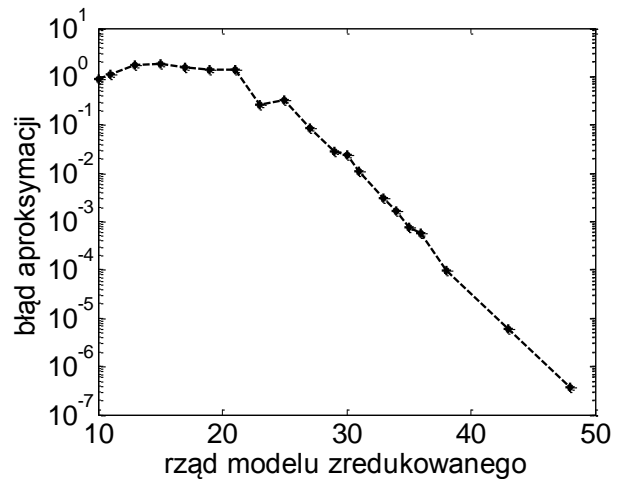
z wykorzystaniem przybornika eAMOR z zakresem adekwatności do 0,4 rad/s. Uzyskano 20 modeli zredukowanych, których rząd wynosi od 4 do 42 zmiennych stanu (wraz z dołączoną częścią niestabilną).

Na rys. 117 i 118 przedstawiono błąd aproksymacji charakterystyk częstotliwościowych parownika dla modeli zredukowanych w funkcji rzędu modeli zredukowanych.

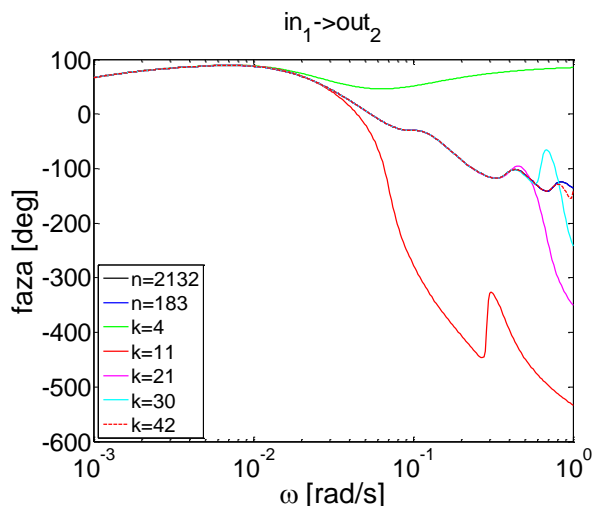
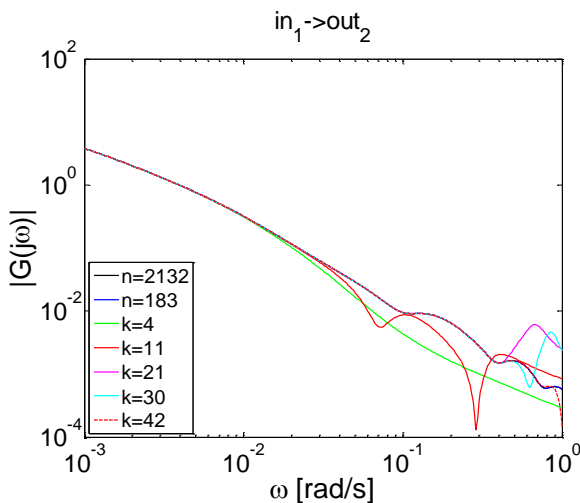
Na rys. 119 - 120 porównano charakterystyki częstotliwościowe modelu parownika kotła BP-1150 otrzymane na podstawie modelu 183 rzędu, 2132 rzędu oraz modeli zredukowanych niskiego rzędu, dla wybranych wielkości wejściowych i wyjściowych.



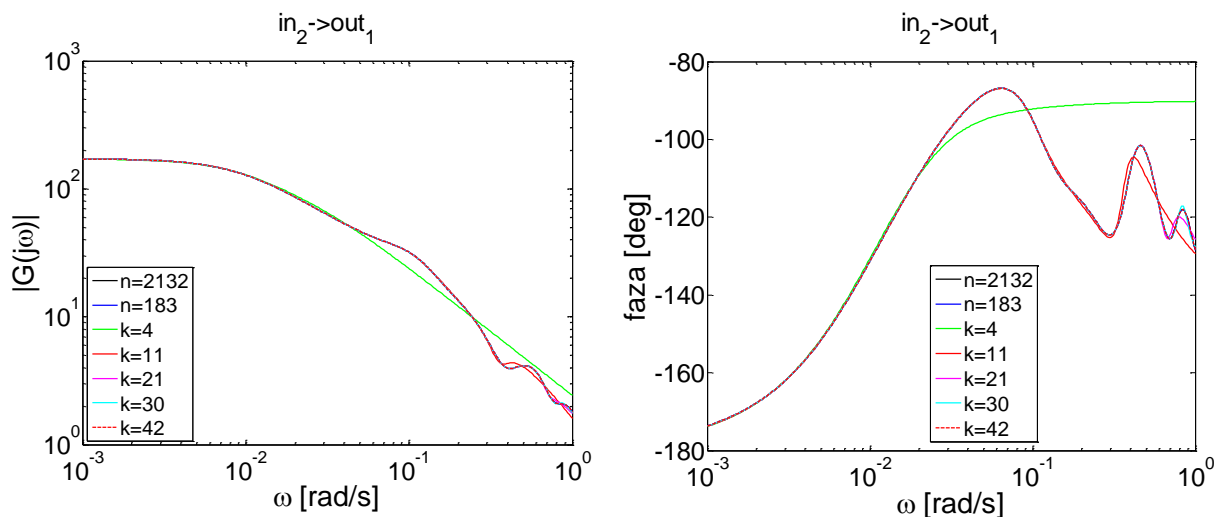
RYSUNEK 117. BŁĄD MODELI ZREDUKOWANYCH PAROWNIKA W STOSUNKU DO MODELU 183 RZĘDU (MODELU ODNIESIENIA REDUKCJI)



RYSUNEK 118. BŁĄD MODELI ZREDUKOWANYCH PAROWNIKA W STOSUNKU DO MODELU WSTĘPNIE ZREDUKOWANEGO RZĘDU 2132 ZMIENNYCH STANU



RYSUNEK 119. PRZYKŁADOWE WYKRESY CHARAKTERYSTYK CZĘSTOTLIWOŚCIOWYCH DLA MODELI PAROWNIKA (tor: $\Delta M_p \rightarrow \Delta H_{sep}$)



RYСУNEK 120 PRZYKŁADOWE WYKRESY CHARAKTERYSTYK CZĘSTOTLIWOŚCIOWYCH DLA MODELI PAROWNIKA (tor: $\Delta q^{\sim} \rightarrow \Delta P_{sep}$)

7.7. Uwagi

W rozdziale przedstawiono metodykę dekompozycji i redukcji złożonych modeli obiektów sterowania na przykładzie redukcji modelu parownika kotła energetycznego BP-1150. Model pierwotny parownika, otrzymany metodą elementów skończonych stosowaną bardzo często do modelowania obiektów o parametrach rozłożonych, charakteryzuje się bardzo wysokim rzędem (około 15 tysięcy zmiennych stanu). Tak wysoki rząd modelu wynika przede wszystkim z rozłożenia parametrów wzdłuż długości rur ekranowych (całkowita długość około 100 m), wzdłuż promienia oraz obwodu ścianek rur ekranowych, a także wzdłuż płetwy łączącej w szczelnej ścianie komory paleniskowej (przewodnictwo cieplne).

W modelu matematycznym analizowanym w pracy opracowano model rur ekranowych na podstawie jednej uśrednionej rury parownika, o średnim przepływie masowym i średnim obciążeniu cieplnym. Aby uzyskać model parownika dokładniej opisujący procesy dynamiczne, należałoby wziąć pod uwagę przynajmniej kilka rur o różnych przepływach czynnika roboczego i różnych obciążeniach cieplnych. Pociągałoby to za sobą znaczną rozbudowę modelu matematycznego, a ogólny rząd modelu przekroczyłby prawdopodobnie 100 000.

Opracowane modele matematyczne parownika charakteryzują się zakresem adekwatności do około 10-20 rad/s, ze względu na przyjęte założenia upraszczające przyjęte podczas tworzenia modelu (np. założenie równowagi termodynamicznej mieszaniny parowo-wodnej). Przy jednokrotnej redukcji modelu parownika (rząd modelu około 15000 zmiennych stanu) zakres aproksymacji charakterystyk częstotliwościowych powinien wynosić około 20 rad/s. Jest to jednak zadanie niewykonalne z wykorzystaniem dostępnego sprzętu komputerowego. Z tego powodu, problem redukcji tak złożonego modelu, postanowiono rozwiązać poprzez dekompozycję modelu na szereg podsystemów, zredukować modele podsystemów, a następnie scalić modele zredukowane. Dekompozycję modeli parownika przeprowadzono na trzech poziomach:

- dekompozycja fizyczna – wydzielenie podsystemów biorąc pod uwagę konstrukcję i funkcjonowanie obiektu,
- dwupoziomowa dekompozycja matematyczna (algebraiczna).

Modele zdekomponowane posiadają rozmiar, który jest akceptowalny z punktu widzenia czasu obliczeń oraz rozmiaru pamięci. Niestety w wyniku scalania modeli zredukowanych

uzyskuje się często zredukowane modele niestabilne. W celu zapewnienia stabilności niezbędny jest szeroki zakres aproksymacji charakterystyk częstotliwościowych przez zdekomponowane modele zredukowane (dla modeli rur ekranowych – 1000 rad/s). Pociąga to za sobą wysoki rząd modeli zredukowanych. Problem ten rozwiązano poprzez wielokrotne redukcje modeli parownika.

Biorąc pod uwagę przebieg charakterystyk częstotliwościowych modelu parownika kotła BP-1150, zakres adekwatności modelu wystarczający do właściwego zaprojektowania układu sterowania wynosi około 0,5 rad/s.

Metody dekompozycji zastosowane do redukcji modeli podsystemów rur ekranowych, wybrane na podstawie analiz zawartych w rozdziale 7.1.2, dawały w zdecydowanej większości zadawalające rezultaty, potwierdzając, że przyjęta metodologia postępowania została dobrana właściwie. Dekompozycja Str/Str w zastosowaniu do modeli podsystemów rur ekranowych, opisanych macierzami rzadkimi, poprzez włączenie interfejsów między wydzielonymi podsystemami do procesu redukcji, charakteryzuje się najmniejszym błędem aproksymacji. Zastosowanie dekompozycji Str/Str wiąże się ze sporadycznym uzyskaniem modelu niestabilnego. Nieco gorsze rezultaty otrzymano z zastosowaniem dekompozycji $Str/SchSylv$. Uzyskane modele zredukowane cechują się błędem aproksymacji często gorszym o 5 rzędów od dekompozycji Str/Str . Również częściej w wyniku redukcji uzyskuje się modele niestabilne. Jednak 65%, ze wszystkich analizowanych, modeli stabilnych oraz niski błąd aproksymacji dla modeli wyższego rzędu pozwala zaliczyć dekompozycję $Str/SchSylv$ do grupy akceptowalnych dekompozycji modeli podsystemów rur ekranowych. Dekompozycje slow-fast: $Esf-SchSylv/SchSylv$, $Esf-SchSylv/SchComp$, $Esf-Sch/SchSylv$ dla każdego analizowanego przypadku dają stabilny model zredukowany, bez względu na wynikowy rząd. W większości przypadków podniesienie rzędu części szybkiej modelu przynosi zdecydowaną poprawę błędu redukcji. Dekompozycja $Sch/SchSylv$ cechuje się najgorszymi właściwościami w stosunku do modeli podsystemów rur ekranowych. Pomimo tego, że 100% uzyskanych modeli w wyniku dekompozycji to modele stabilne, błąd aproksymacji dla większości otrzymanych modeli jest niedopuszczalny. Wysoki błąd metody wynika ze struktury modelu zredukowanego. Pomimo niskiego błędu aproksymacji modeli podsystemów, model scalony posiada nieproporcjonalnie gorsze właściwości.

8. Weryfikacja modeli zredukowanych

Modele zredukowane parownika kotła BP-1150 uzyskano na podstawie złożonych obliczeń z zastosowaniem dekompozycji oraz szeregu operacji redukcji rzędu modeli podsystemów, z zastosowaniem metody FW oraz z zastosowaniem obliczeń równoległych. Metoda redukcji FW wykorzystuje algorytmy genetyczne do wyznaczenia optymalnych parametrów redukcji oraz z wykorzystaniem obliczeń równoległych. Tak uzyskany model niskiego rzędu postanowiono poddać weryfikacji w celu potwierdzenia poprawności realizowanych operacji redukcji rzędu modelu. Takiej weryfikacji można dokonać poprzez wykorzystanie uzyskanych modeli zredukowanych do zaprojektowania układu sterowania parownika kotła BP-1150, a następnie zastosowanie opracowanego układu sterowania do rzeczywistego obiektu.

Uzyskane modele można poddać weryfikacji na drodze eksperymentalnej jedynie w bardzo ograniczonym zakresie. Wynika to przede wszystkim z faktu, że ingerencja w pracę bloku energetycznego podczas normalnej eksploatacji może mieć bardzo ograniczony charakter. Z tego powodu pozostaje się jedynie weryfikacja modelu zredukowanego poprzez:

- projekt układu sterowania na podstawie modelu zredukowanego,
- zastosowanie opracowanego regulatora do modelu pierwotnego (wysokiego rzędu).

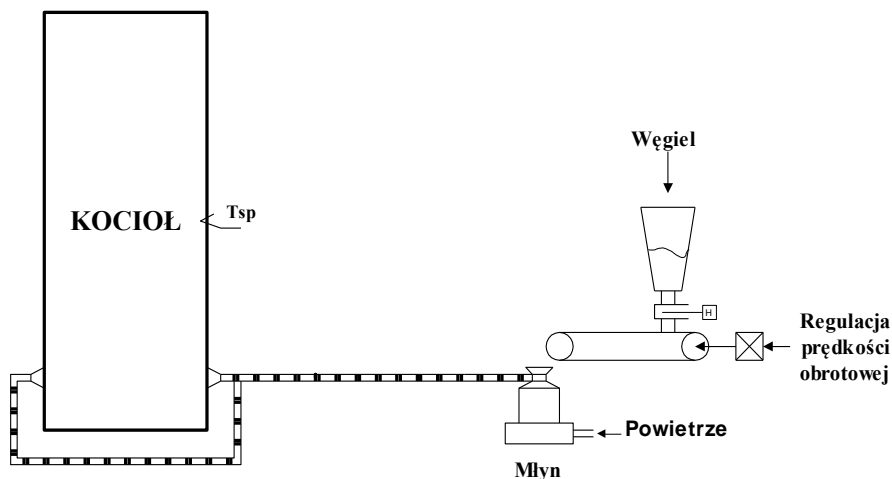
Model pierwotny parownika przepływowego zawiera około 15 tysięcy zmiennych stanu, co powoduje, że wyznaczenie charakterystyk dla takiego modelu jest praktycznie niemożliwe ze względów numerycznych. Z tego powodu, jako model odniesienia, przyjęto model wstępnie zredukowany, zawierający następujące modele:

- rur ekranowych – 1448 zmiennych stanu (model wstępnie zredukowany),
- separatora – 72 zmiennych stanu (model oryginalny),
- układu cyrkulacji – 612 zmiennych stanu (model oryginalny),
- układu przygotowania paliwa – 6 zmiennych stanu (model oryginalny).

8.1. Układ regulacji ciśnienia w kotle BP-1150

W parowniku kotła BP-1150 występują dwie wielkości sterujące \tilde{q} , M_{zas} oraz szereg wielkości zakłócających, mających istotny wpływ na zmiany wielkości wyjściowych: ciśnienia P_{sep} oraz poziomu wody H_{sep} w separatorze.

Względna wartość strumienia cieplnego \tilde{q} doprowadzanego do rur ekranowych parownika wynika przede wszystkim ze zmian ilości paliwa (węгля) dostarczanego do młynów węglowych, dokonywanych poprzez zmiany szybkości transporterów taśmowych węгля. W młynach węglowych węgiel mielony jest na pył węglowy i w postaci mieszaniny pyłowo-powietrznej transportowany jest do palników, umieszczonych w czterech narożach kotła [32]. Uproszczony schemat technologiczny układu przygotowania paliwa przedstawiono na rys. 121.



RYSUNEK 121. UPROSZCZONY SCHEMAT TECHNOLOGICZNY UKŁADU PRZYGOTOWANIA PALIWA

Kotły BP-1150 zainstalowane w Elektrowni „OPOLE” S.A. wyposażone są w sześć zespołów młynowych misowo-rolkowych. Ich zadaniem jest rozdrobnienie węgla kamiennego, suszenie i podanie go jako mieszanki pyłowo-powietrznej poprzez palniki pyłowe do komory paleniskowej kotła. W skład każdego zespołu młynowego wchodzi: młyn węglowy wraz z przekładnią, wentylator młynowy, podajnik węgla, zasobnik węgla, kanały gorącego i zimnego powietrza, przewody pyłu węglowego. Do pracy przy pełnym obciążeniu kotła BP-1150 potrzebna jest praca czterech zespołów młynowych. Ponieważ zmiana ilości paliwa dopływającego do komory paleniskowej kotła odbywa się poprzez zmianę prędkości podajnika, układ przygotowania paliwa charakteryzuje się znacznym opóźnieniem ($\tau \cong 50$ sek.) oraz stałą czasową ($T \cong 300$ sek.). Transmitancja układu przygotowania paliwa zgodnie z [150] ma następującą postać: $\frac{q(s)}{OBR.POD\%(s)} = \frac{k}{Ts + 1} e^{-\tau s}$, gdzie OBR.POD% – prędkość obrotowa podajników węgla mierzona w %.

Do projektowania układu sterowania kotła przyjęto, że model układu przygotowania paliwa posiada transmitancję $G_{UPP}(s) = \frac{k}{(Ts + 1)(\tau/5 \cdot s + 1)^5}$.

Ciśnienie oraz poziom wody w separatorze są miarami aktualnego stanu bilansu energii cieplnej i masy w kotle, a zmiany ciśnienia i poziomu wody w separatorze świadczą o aktualnym braku równowagi w energii cieplnej dopływającej do parownika i energii cieplnej strugi pary wypływającej z separatora, oraz masy wody dopływającej do parownika i masy pary wypływającej z separatora. Układ regulacji ciśnienia w kotle zapewnia bilansowanie energii cieplnej w parowniku, poprzez oddziaływanie na ilość paliwa dostarczanego do kotła. Układ ten jednak nie jest w stanie zapewnić stałego ciśnienia w separatorze przy szybkich i silnych wahaniami poboru pary z kotła ze względu na znaczne opóźnienie wprowadzane przez układ przygotowania paliwa [150].

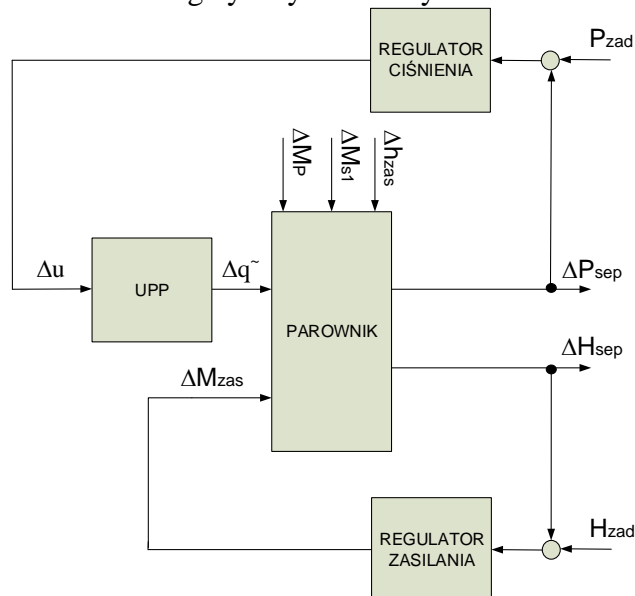
Głównymi zakłóceniami oddziałującymi na parownik są: zmiany poboru pary z separatora M_p , zmiany entalpii wody zasilającej parownik h_{zas} , oraz zmiany poboru wody do schładzaczy pary M_{sl} . Najistotniejsze znaczenie dla pracy układu regulacji ciśnienia mają zmiany poboru pary z separatora M_p spowodowane zmianami stopnia otwarcia zaworów regulacyjnych turbiny. Zawory regulacyjne turbiny są organem wykonawczym w układzie regulacji prędkości obrotowej turbiny i przestawiane są w wyniku zmian częstotliwości w systemie energetycznym (regulacja pierwotna) oraz w wyniku zmian sygnałów wartości zadanej mocy turbozespołu, dostarczanych z centralnego regulatora mocy (regulacja wtórna).

Zmiany przepływu masowego pary na wylocie z kotła M_p są bardzo szybkie i mogą mieć duże wartości. Czerpanie większej ilości pary z kotła niż to wynika z aktualnego bilansu energii oraz masy w parowniku jest możliwe w krótkim przedziale czasu dzięki zdolności akumulacyjnej kotła. Niestety w kotłach przepływowych BP-1150 zdolność akumulacyjna nie jest wysoka ze względu na niewielką objętość wodno-parową parownika (separator ma niewielką pojemność w stosunku do walczaka stosowanego w kotłach klasycznych). Jednocześnie aktualne wymagania co do wahań poboru pary z kotła wzrastają ze względu na udział bloków energetycznych w pierwotnej i wtórnej regulacji mocy i częstotliwości. Zmiany przepływu pary M_p powodują znaczne wahania ciśnienia pary w separatorze, gdyż układ regulacji ciśnienia nie jest w stanie kompensować szybkich zakłóceń ze względu na znaczne opóźnienie oraz stałą czasową w układzie przygotowania paliwa. Wprowadza to znaczne ograniczenia na wielkość i szybkość zmian poboru pary z separatora, co powoduje, że nie można w pełni wykorzystać zdolności akumulacyjnej kotła [150].

Zmiany przepływu wody do schładzaczy pary M_{s1} wynikające z funkcjonowania układu regulacji temperatury pary świeżej, powodują przede wszystkim zmiany poziomu wody w separatorze, gdyż naruszają bilans masy w parowniku. Wpływ zmian przepływu M_{s1} na ciśnienie w separatorze jest niewielki [150].

Zmiany entalpii wody zasilającej h_{zas} mają istotny wpływ na zmiany ciśnienia a także na zmiany poziomu wody w separatorze. Biorąc jednak pod uwagę, że parownik jest zasilany wodą z podgrzewacza wody (ekonomizer), charakteryzującego się znaczną stałą czasową, zmiany entalpii wody zasilającej są bardzo powolne. Powoduje to, że wpływ tego zakłócenia na pracę parownika jest niewielki [150].

Na rys. 122 przedstawiono schemat układu regulacji parownika kotła BP-1150 aktualnie stosowany w blokach energetycznych o mocy 300 MW.

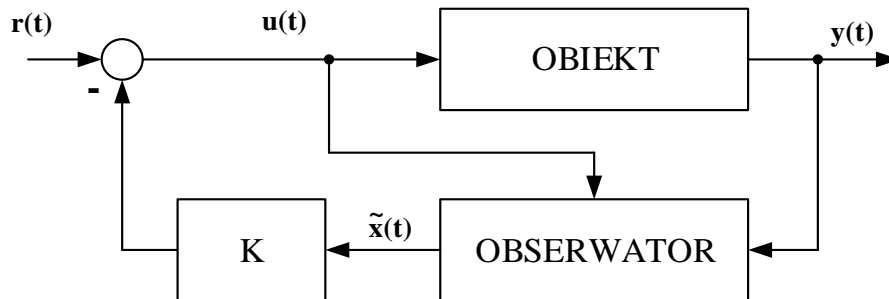


RYSUNEK 122. SCHEMAT UKŁADU REGULACJI PAROWNIKA KOTŁA BP-1150

Widmo zakłóceń przepływem pary M_p sięga częstotliwości $\omega_{max}=10^{-2}$ rad/s. Określa to warunki pracy układu regulacji ciśnienia w kotle. Zakłócenie entalpią wody zasilającej h_{zas} posiada ok. 10-krotnie mniejszą częstotliwość, oraz znacznie mniejszą amplitudę.

8.2. Układ regulacji

Ogólny schemat obiektu połączonego z obserwatorem wektora stanu oraz sprzężeniem zwrotnym został przedstawiony na rysunku 123 [151]:



RYСУNEK 123. SCHEMAT OBIEKTU Z OBSERWATOREM STANU I SPRZEŻENIEM ZWROTNYM

Równania opisujące układ z rysunku 123 wyrażone są następująco [151]:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\tilde{x}}(t) \end{bmatrix} = \begin{bmatrix} A & -BK \\ LC & A-LC-BK \end{bmatrix} \begin{bmatrix} x(t) \\ \tilde{x}(t) \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} r(t) \quad (142)$$

$$y(t) = [C \quad -DK] \begin{bmatrix} x(t) \\ \tilde{x}(t) \end{bmatrix} + Dr(t) \quad (143)$$

Wyznaczony model posiada dwa razy większy rozmiar od rozmiaru modelu obiektu. Wykorzystując macierz przekształcenia równoważnościowego [151]:

$$P = P^{-1} = \begin{bmatrix} I & \emptyset \\ I & -I \end{bmatrix} \quad (144)$$

równania (142) i (143) można doprowadzić do postaci [151]:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{e}(t) \end{bmatrix} = \begin{bmatrix} A-BK & BK \\ \emptyset & A-LC \end{bmatrix} \begin{bmatrix} x(t) \\ e(t) \end{bmatrix} + \begin{bmatrix} B \\ \emptyset \end{bmatrix} r(t) \quad (145)$$

$$y(t) = [C-DK \quad DK] \begin{bmatrix} x(t) \\ e(t) \end{bmatrix} + Dr(t) \quad (146)$$

8.3. Regulator LQG

Równania stanu systemu, dla którego projektowany jest układ sterowania LQG wyrażone są zależnościami [152]:

$$\dot{x}(t) = Ax(t) + Bu(t) + \Gamma w(t) \quad (147)$$

$$y(t) = Cx(t) + Du(t) + v(t) \quad (148)$$

Równania wprowadzają zakłócenia wpływające na wartości zmiennych stanu (szum procesu) oraz na wartości wyjściowe (szum pomiaru). Zakłócenia $w(t)$ oraz $v(t)$ posiadają następujące cechy [153]:

- szum procesu $w(t)$ jest nieskorelowany oraz posiada zerową wartość oczekiwaną:

$$E[w(t)] = 0, \quad E[w(t)w^T(\tau)] = 0, \quad t \neq \tau \quad (149)$$

- macierz kowariancji szumu procesu jest symetryczna, dodatnio określona:

$$E[w(t)w^T(\tau)] = W\delta(t-\tau) \quad (150)$$

- szum pomiaru $v(t)$ jest nieskorelowany oraz posiada zerową wartość oczekiwaną:

$$E[v(t)] = 0, E[v(t)v^T(\tau)] = 0, t \neq \tau \quad (151)$$

- macierz kowariancji szumu pomiaru jest symetryczna, dodatnio określona:

$$E[v(t)v^T(\tau)] = V\delta(t-\tau) \quad (152)$$

- szum procesu oraz szum pomiaru są wzajemnie nieskorelowane:

$$E[v(t)w^T(\tau)] = 0, E[w(t)v^T(\tau)] = 0 \quad (153)$$

funkcja $\delta(t-\tau)$ określa deltę Kroneckera.

Synteza układu sterowania związana jest z minimalizacją wskaźnika jakości, wyrażonego wzorem [152]:

$$J = \lim_{T \rightarrow \infty} \frac{1}{T} E \left\{ \int_0^T [x^T(t)Qx(t) + u^T(t)Ru(t)] dt \right\} \quad (154)$$

gdzie macierz Q jest macierzą symetryczną, nieujemnie określoną, natomiast macierz R jest macierzą symetryczną, dodatnio określoną. Rozwiązanie problemu syntezy regulatora metodą LQG sprowadza się do rozwiązania dwóch niezależnych problemów. Pierwszy z nich związany jest z wyznaczeniem macierzy sprzężenia zwrotnego K_r dla prawa sterowania [152]:

$$u(t) = -K_r x(t) \quad (155)$$

W następnym kroku wyznaczany jest estymator wektora stanu, tak aby zminimalizować wartość wyrażenia:

$$E \left\{ (x - \hat{x})^T (x - \hat{x}) \right\} \quad (156)$$

gdzie \hat{x} określa estymowaną wartość wektora stanu. Estymator wektora stanu (obserwator stanu) realizowany jest jako filtr Kalmana. Optymalny dobór macierzy wzmocnień błędu wyjściowego realizowany jest zgodnie z zależnością [152]:

$$L_{KALMAN} = YC^T V^{-1} \quad (157)$$

gdzie macierz Y jest macierzą symetryczną, nieujemnie określoną wyznaczaną z równania [152]:

$$YA^T + AY - YC^T V^{-1} CY + W = 0 \quad (158)$$

Wyznaczanie macierzy sprzężenia zwrotnego realizowane jest metodą LQR (ang. Linear Quadratic Regulator). Rozwiązanie problemu polega na minimalizacji funkcji jakości [152]:

$$J_{LQR} = \int_0^{\infty} [x^T(t)Qx(t) + u^T(t)Ru(t)] dt \quad (159)$$

Optymalne rozwiązanie wyrażone jest wzorem [152]:

$$K_{LQR} = R^{-1} B^T X \quad (160)$$

gdzie $X = X^T \geq 0$ jest rozwiązaniem równania Riccati'ego [152]:

$$A^T X + XA - XBR^{-1}B^T X + Q = 0 \quad (161)$$

8.4. Projekt regulatora na podstawie zredukowanego modelu parownika

Ze względu na brak możliwości analizy układu sterowania dla modelu oryginalnego – rząd około 15 tysięcy zmiennych stanu, za model odniesienia przyjęto model wstępnie zredukowany. Układy regulacji wyznaczono z wykorzystaniem zredukowanych modeli parownika, z dołączonym układem przygotowania paliwa (6 zmiennych stanu), których rzędy wynosiły od 10 do 33 zmiennych stanu. Dla modeli wyższego rzędu, wyznaczanie układów regulacji z wykorzystaniem wbudowanych poleceń środowiska MATLAB okazało się

niemożliwe. Wykorzystano cztery podstawowe metody projektowania układu regulacji z obserwatorem stanu dla modeli MIMO:

- LQG – obserwator stanu Kalmana, macierz wzmocnień regulatora wyznaczona metodą LQRY,
- PLACE-LQRY – macierz wzmocnień błędu wyjściowego wyznaczona metodą lokowania wartości własnych, macierz wzmocnień regulatora wyznaczona algorytmem LQRY,
- KALMAN-PLACE – obserwator stanu Kalmana, macierz wzmocnień regulatora wyznaczona metodą lokowania wartości własnych,
- PLACE-PLACE – zastosowanie metody lokowania wartości własnych do wyznaczenia macierzy wzmocnień obserwatora stanu i regulatora.

Dla metody LQG wyznaczono 45 układów sterowania dla każdego rozważanego modelu zredukowanego parownika z dołączonym układem przygotowania paliwa. Elementami podlegającymi modyfikacji przy wyznaczaniu układu sterowania dla pojedynczego modelu były macierz kowariancji szumu pomiaru V (dla obserwatora Kalmana) oraz macierz wag wyjściowych R (dla metody LQRY). Obie macierze stanowią macierze diagonalne, z wartościami na przekątnej równymi r_o dla macierzy V oraz r_R dla macierzy R . Wartość parametru r_o modyfikowano począwszy od 10^{-6} do $6,56 \cdot 10^{-4}$ (gdzie kolejna wartość parametru r_o wynosi $r_o^{k+1} = 9 \cdot r_o^k$). Wartość parametru r_R modyfikowano począwszy od 10^{-7} do $6,56 \cdot 10^{-3}$ (gdzie kolejna wartość r_R wynosi $r_R^{k+1} = 3 \cdot r_o^k$). Macierze W (dla obserwatora Kalmana) oraz Q (dla metody LQRY) przyjęto jako macierze jednostkowe.

Dla metody PLACE-LQRY wyznaczono 64 układów sterowania dla każdego rozważanego modelu. Elementami podlegającymi modyfikacji przy wyznaczaniu układu sterowania dla pojedynczego modelu są macierz R dla metody LQRY oraz wektor zadanych wartości własnych dla metody lokowania biegunów (PLACE). Wartość parametru r_R modyfikowano począwszy od 10^{-4} do $2,19 \cdot 10^{-1}$. W przypadku metody LQRY pozostałe założenia wyglądają analogicznie jak dla metody LQG. Wektor zadanych wartości własnych dla metody PLACE wyznaczany został z zależności: $D = r_o \cdot \lambda_{minus}$, gdzie zapis λ_{minus} oznacza wektor wartości własnych modelu oryginalnego ze wszystkimi częściami rzeczywistymi ujemnymi. Parametr r_o modyfikowano od wartości 1 do 1,7 z krokiem 0,1.

Dla metody KALMAN-PLACE wyznaczono 96 układów sterowania dla każdego rozważanego modelu. Wartość parametru r_o (analogiczne znaczenia jak dla metody LQG) modyfikowano począwszy od 10^{-4} do 0,94 (gdzie kolejna wartość parametru r_o wynosi $r_o^{k+1} = 3,7 \cdot r_o^k$). Wartość parametru r_R dla metody lokowania biegunów (analogiczne znaczenie jak dla parametru r_o w metodzie PLACE-LQRY) modyfikowano od 1 do 2,1 z krokiem 0,1.

Dla metody PLACE-PLACE wyznaczono 81 układów sterowania dla każdego rozważanego modelu. Wartość parametru r_o modyfikowano od 1 do 1,8 z krokiem 0,1. Wartość parametru r_R modyfikowano od 1 do 2,36 z krokiem 0,17.

W tabeli 10 zamieszczono podstawowe parametry wybranych układów sterowania po zastosowaniu ich do modelu wysokiego rzędu. Parametr $SR_{1,1}$ oznacza średnią wartość modułu transmitancji dla toru $\Delta M_p \rightarrow \Delta P_{SEP}$ dla przedziału częstotliwości $\langle 10^{-4} - 10^{-2} \text{ rad/s} \rangle$ dla układu zamkniętego. Parametr $SR_{1,2}$ oznacza średnią wartość modułu transmitancji dla toru $\Delta M_p \rightarrow \Delta H_{SEP}$ dla przedziału częstotliwości $\langle 10^{-4} - 10^{-2} \text{ rad/s} \rangle$ dla układu zamkniętego. Parametr $GM_{2,1}$ oznacza zapas amplitudy (w dB) dla toru sterowania $\Delta U_{UPP} \rightarrow \Delta P_{SEP}$ wyznaczony dla układu otwartego. Parametr

PM_{2_1} oznacza zapas fazy (w stopniach) dla toru sterowania $\Delta U_{UPP} \rightarrow \Delta P_{SEP}$ wyznaczony dla układu otwartego.

TABELA 10. PODSTAWOWE PARAMETRY WYZNACZONYCH UKŁADÓW STEROWANIA

Metoda Rząd modelu	LQG	PLACE-LQRY	KALMAN-PLACE	PLACE-PLACE
10	SR _{1_1} =0,24 SR _{1_2} =0,02 GM _{2_1} =3,23 PM _{2_1} =38,78 r ₀ =6,56E-04, r _R =6,56E-03	SR _{1_1} =0,15 SR _{1_2} =0,03 GM _{2_1} =9,41 PM _{2_1} =40,95 r ₀ =1, r _R =2,70E-03	-	-
11	SR _{1_1} =0,24 SR _{1_2} =0,03 GM _{2_1} =3,14 PM _{2_1} =37,78 r ₀ =6,56E-04, r _R =6,56E-03	SR _{1_1} =0,26 SR _{1_2} =0,02 GM _{2_1} =11,88 PM _{2_1} =24,61 r ₀ =1, r _R =2,43E-02	SR _{1_1} =16,52 SR _{1_2} =17,98 GM _{2_1} =0,43 PM _{2_1} =3,56 r ₀ =2,57E-01, r _R =1,1	SR _{1_1} =3,81 SR _{1_2} =8,05 GM _{2_1} =29,01 PM _{2_1} =14,49 r ₀ =1, r _R =1,85
13	SR _{1_1} =0,2 SR _{1_2} =0,05 GM _{2_1} =2,68 PM _{2_1} =25,61 r ₀ =6,56E-04, r _R =6,56E-03	SR _{1_1} =0,53 SR _{1_2} =0,25 GM _{2_1} =9,63 PM _{2_1} =34,64 r ₀ =1, r _R =2,43E-02	SR _{1_1} =5,49 SR _{1_2} =1,03 GM _{2_1} =11,62 PM _{2_1} =4,97 r ₀ =1,00E-04, r _R =2,1	SR _{1_1} =4,49 SR _{1_2} =0,46 GM _{2_1} =17,78 PM _{2_1} =8,32 r ₀ =1,2, r _R =2,36
15	SR _{1_1} =0,21 SR _{1_2} =0,07 GM _{2_1} =1,86 PM _{2_1} =18,42 r ₀ =6,56E-04, r _R =6,56E-03	SR _{1_1} =0,34 SR _{1_2} =0,09 GM _{2_1} =10,33 PM _{2_1} =26,73 r ₀ =1, r _R =8,10E-03	SR _{1_1} =4,33 SR _{1_2} =1,87 GM _{2_1} =13,36 PM _{2_1} =9,48 r ₀ =3,70E-04, r _R =2	SR _{1_1} =2,47 SR _{1_2} =5,63 GM _{2_1} =73,06 PM _{2_1} =4496 r ₀ =1,6, r _R =1,68
17	SR _{1_1} =0,12 SR _{1_2} =0,02 GM _{2_1} =6,83 PM _{2_1} =39,06 r ₀ =6,56E-04, r _R =1,00E-06	SR _{1_1} =0,44 SR _{1_2} =0,05 GM _{2_1} =10,22 PM _{2_1} =30,64 r ₀ =1, r _R =2,43E-02	SR _{1_1} =4,06 SR _{1_2} =0,83 GM _{2_1} =13,68 PM _{2_1} =15,57 r ₀ =1,00E-04, r _R =2,1	SR _{1_1} =5,57 SR _{1_2} =1,76 GM _{2_1} =8,57 PM _{2_1} =5,45 r ₀ =1,4, r _R =1,51
19	SR _{1_1} =0,13 SR _{1_2} =0,03 GM _{2_1} =2,88 PM _{2_1} =20,45 r ₀ =6,56E-04, r _R =1,00E-06	SR _{1_1} =0,25 SR _{1_2} =0,04 GM _{2_1} =10,24 PM _{2_1} =26,02 r ₀ =1, r _R =8,10E-03	SR _{1_1} =4,45 SR _{1_2} =4,2 GM _{2_1} =6,25 PM _{2_1} =20 r ₀ =9,49E-01, r _R =2,1	SR _{1_1} =1,58 SR _{1_2} =1,57 GM _{2_1} =11,17 PM _{2_1} =31,61 r ₀ =1,1, r _R =2,19
21	SR _{1_1} =0,25 SR _{1_2} =0,04 GM _{2_1} =2,17 PM _{2_1} =21,64 r ₀ =6,56E-04, r _R =6,56E-03	SR _{1_1} =0,28 SR _{1_2} =0,01 GM _{2_1} =11,04 PM _{2_1} =20,92 r ₀ =1, r _R =2,43E-02	SR _{1_1} =4,69 SR _{1_2} =1,32 GM _{2_1} =5,8 PM _{2_1} =14,51 r ₀ =2,57E-01, r _R =2	SR _{1_1} =1,7 SR _{1_2} =0,4 GM _{2_1} =12,53 PM _{2_1} =33,71 r ₀ =1,1, r _R =2,36
23	SR _{1_1} =0,24 SR _{1_2} =0,04 GM _{2_1} =2,28 PM _{2_1} =21,38 r ₀ =6,56E-04, r _R =6,56E-03	SR _{1_1} =0,25 SR _{1_2} =0,01 GM _{2_1} =7,35 PM _{2_1} =28,71 r ₀ =1,6, r _R =2,43E-02	SR _{1_1} =1,53 SR _{1_2} =2,69 GM _{2_1} =7,24 PM _{2_1} =12,25 r ₀ =1,37E-03, r _R =1	SR _{1_1} =5,36 SR _{1_2} =1,32 GM _{2_1} =11,52 PM _{2_1} =5,76 r ₀ =1,3, r _R =1,34
29	SR _{1_1} =0,24 SR _{1_2} =0,04 GM _{2_1} =2,2 PM _{2_1} =21,03 r ₀ =6,56E-04, r _R =6,56E-03	SR _{1_1} =0,21 SR _{1_2} =0,04 GM _{2_1} =11,17 PM _{2_1} =21,19 r ₀ =1, r _R =8,10E-03	SR _{1_1} =3,35 SR _{1_2} =0,49 GM _{2_1} =7,75 PM _{2_1} =20,85 r ₀ =9,49E-01, r _R =1,8	SR _{1_1} =3,33 SR _{1_2} =0,49 GM _{2_1} =9,64 PM _{2_1} =49,7 r ₀ =1, r _R =1,51
31	SR _{1_1} =0,24 SR _{1_2} =0,04 GM _{2_1} =2,2 PM _{2_1} =21 r ₀ =6,56E-04, r _R =6,56E-03	SR _{1_1} =0,32 SR _{1_2} =0,08 GM _{2_1} =6,8 PM _{2_1} =33,06 r ₀ =1,4, r _R =7,29E-02	SR _{1_1} =2,68 SR _{1_2} =0,55 GM _{2_1} =7,77 PM _{2_1} =16,69 r ₀ =2,57E-01, r _R =1,8	SR _{1_1} =4,79 SR _{1_2} =0,72 GM _{2_1} =3,89 PM _{2_1} =11,33 r ₀ =1,1, r _R =1,34
33	SR _{1_1} =0,25 SR _{1_2} =0,03 GM _{2_1} =2,35 PM _{2_1} =22,42 r ₀ =6,56E-04, r _R =6,56E-03	SR _{1_1} =0,24 SR _{1_2} =0,02 GM _{2_1} =6,28 PM _{2_1} =31,4 r ₀ =1,6, r _R =2,43E-02	SR _{1_1} =6,35 SR _{1_2} =1,7 GM _{2_1} =6,79 PM _{2_1} =9,25 r ₀ =1,87E-02, r _R =1,3	SR _{1_1} =4,75 SR _{1_2} =1,23 GM _{2_1} =4,77 PM _{2_1} =20,67 r ₀ =1, r _R =1,34

Najlepsze rezultaty osiągnięto dla układów sterowania wyznaczonych metodami LQG oraz PLACE-LQRY.

W przypadku regulatora LQG, najbardziej zadawalający układ sterowania został wyznaczony na podstawie modelu siedemnastego rzędu. Wyznaczony regulator cechuje się wysokim tłumieniem zakłócenia dla torów $\Delta M_p \rightarrow \Delta P_{SEP}$ oraz $\Delta M_p \rightarrow \Delta H_{SEP}$ - średnie wartości modułu transmitancji układu zamkniętego dla przedziału częstotliwości $\langle 10^{-4} - 10^{-2} \text{ rad/s} \rangle$ wynoszą odpowiednio 0,12 oraz 0,02. Wyznaczony regulator spełnia również kryteria zapasu stabilności dla toru regulacji $\Delta U_{UPP} \rightarrow \Delta P_{SEP}$ - zapas amplitudy powyżej 6 dB oraz zapas fazy powyżej 39 stopni.

Dla metody PLACE-LQRY, najbardziej zadawalający układ sterowania został wyznaczony na podstawie modelu dziesiątego rzędu. Wyznaczony regulator posiada nieco gorsze właściwości tłumienia zakłócenia ΔM_p - dla torów $\Delta M_p \rightarrow \Delta P_{SEP}$ oraz $\Delta M_p \rightarrow \Delta H_{SEP}$ - średnie wartości modułu transmitancji układu zamkniętego dla przedziału częstotliwości $\langle 10^{-4} - 10^{-2} \text{ rad/s} \rangle$ wynoszą odpowiednio 0,15 oraz 0,03. W porównaniu do

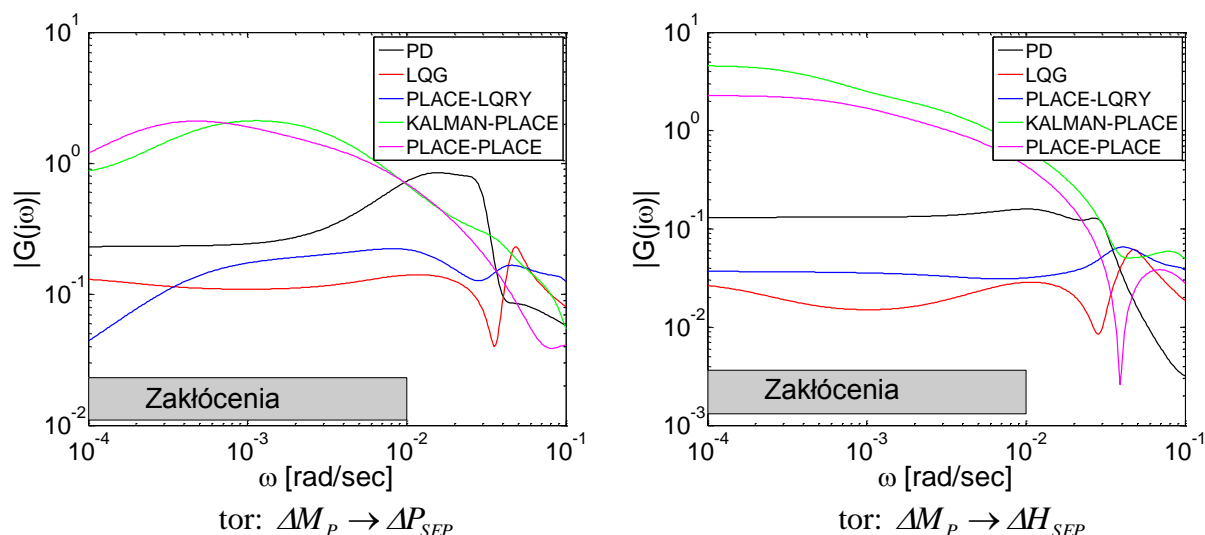
układu regulacji wyznaczonego metodą LQG uzyskano nieco wyższy zapas stabilności dla toru regulacji $\Delta U_{UPP} \rightarrow \Delta P_{SEP}$ - zapas amplitudy powyżej 9 dB, zapas fazy powyżej 40 stopni.

Układy regulacji wyznaczone metodami KALMAN-PLACE oraz PLACE-PLACE, pomimo dobrych parametrów związanych z zapasem stabilności dla toru regulacji $\Delta U_{UPP} \rightarrow \Delta P_{SEP}$, znacznie gorzej tłumią zakłócenie ΔM_p .

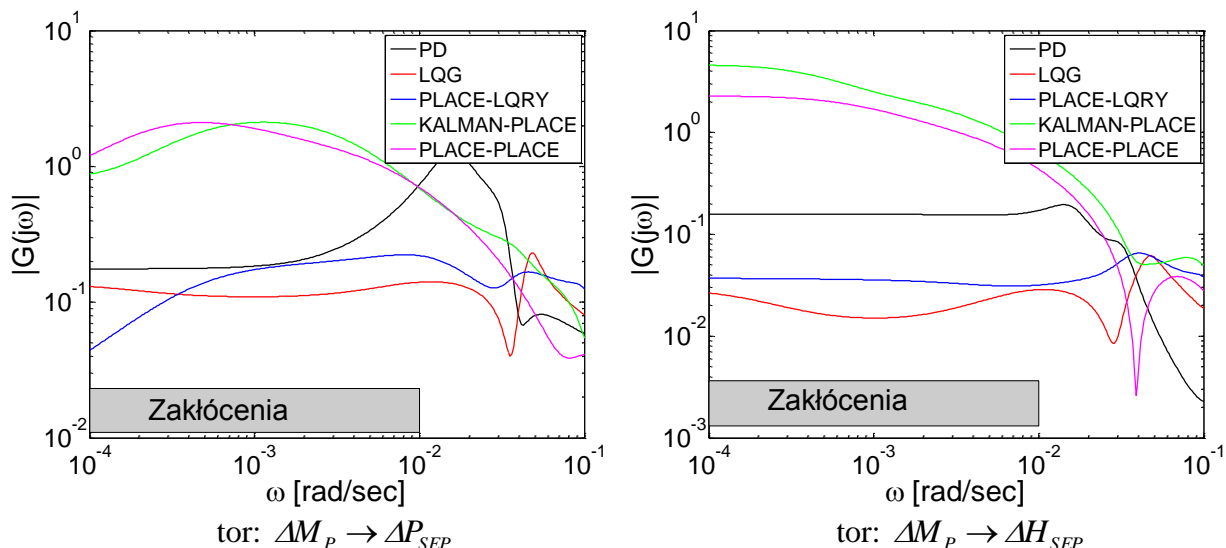
Na rysunkach 124 - 126 porównano wykresy modułu transmitancji układu zamkniętego, przedstawiające stopień tłumienia zakłócenia ΔM_p z zastosowaniem układów regulacji oznaczonych kolorem żółtym w tabeli 10 oraz z klasycznymi regulatorami PD, wyznaczonymi z zastosowaniem modeli rzędu 10, 15 oraz 21 dla torów $\Delta M_p \rightarrow \Delta P_{SEP}$ oraz $\Delta M_p \rightarrow \Delta H_{SEP}$ w zakresie częstotliwości $\langle 10^{-4} - 10^{-1} \text{ rad/s} \rangle$. Dobór nastaw regulatorów PD został wykonany z wykorzystaniem algorytmu ewolucyjnego [154]. Wyznaczone wartości nastaw zostały zamieszczone w tabeli 11.

TABELA 11. DOBÓR NASTAW REGULATORÓW PD DLA MODELI RZĘDU 10, 15, 21

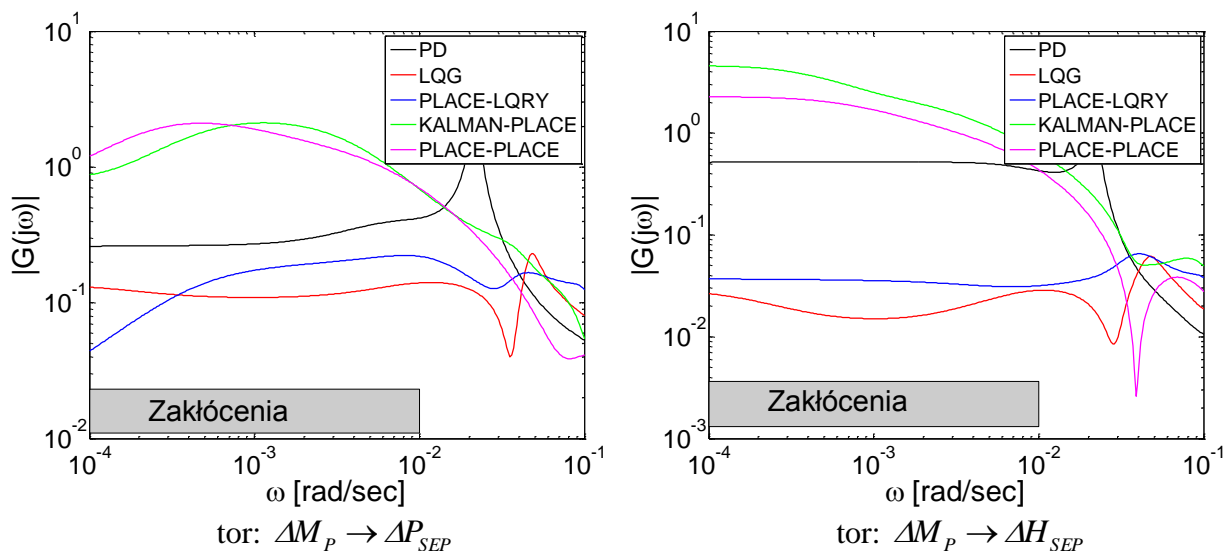
rzęd modelu tor regulacji	k=10	k=15	k=21
PD ₁ - $\Delta U_{UPP} \rightarrow \Delta P_{SEP}$	$k_{p1} = 0.0204$ $T_{d1} = 146,8$	$k_{p1} = 0,0251$ $T_{d1} = 112,9$	$k_{p1} = 0,0188$ $T_{d1} = 186,6848$
PD ₂ - $\Delta M_{zas} \rightarrow \Delta H_{SEP}$	$k_{p2} = 7,7935$ $T_{d2} = 51,09$	$k_{p2} = 6,5183$ $T_{d2} = 95,2153$	$k_{p2} = 2,0488$ $T_{d2} = 5,4846$



RYSUNEK 124. PORÓWNANIE MODUŁU TRANSMITANCJI UKŁADU ZAMKNIĘTEGO (STOPIEŃ TŁUMIENIA ZAKŁÓCENIA ΔM_p) DLA WYBRANYCH REGULATORÓW WYZNACZONYCH METODAMI: LQG, PLACE-LQRY, KALMAN-PLACE, PLACE-PLACE ORAZ PD WYZNACZONYM DLA MODELU RZĘDU 10

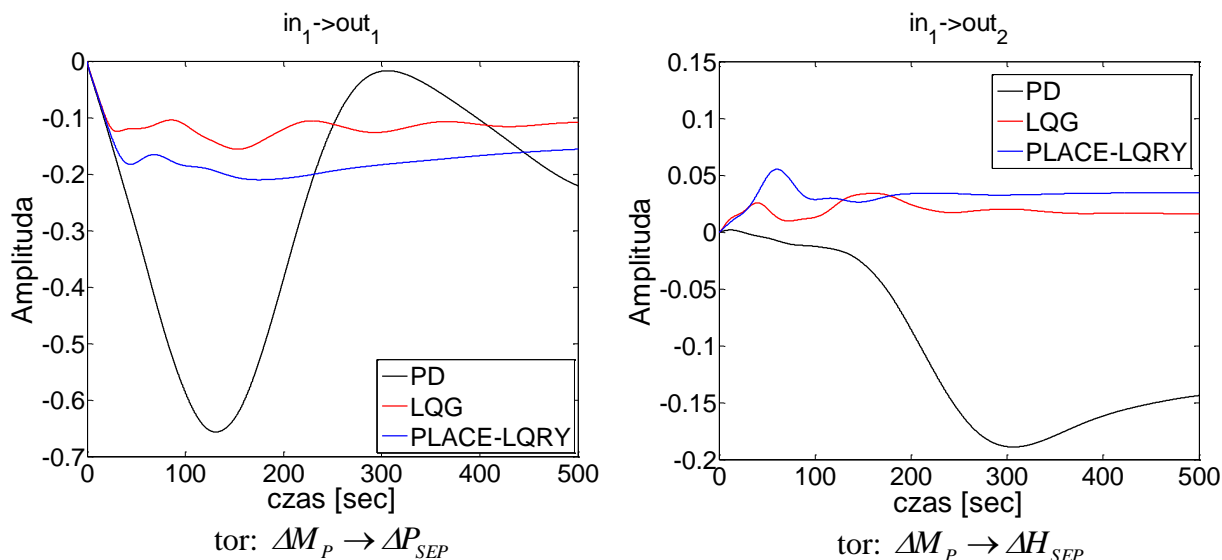


RYSUNEK 125. PORÓWNANIE MODUŁU TRANSMITANCJI UKŁADU ZAMKNIĘTEGO (STOPIEŃ TŁUMIENIA ZAKŁÓCENIA ΔM_p) DLA WYBRANYCH REGULATORÓW WYZNACZONYCH METODAMI: LQG, PLACE-LQRY, KALMAN-PLACE, PLACE-PLACE ORAZ PD WYZNACZONYM DLA MODELU RZĘDU 15



RYSUNEK 126. PORÓWNANIE MODUŁU TRANSMITANCJI UKŁADU ZAMKNIĘTEGO (STOPIEŃ TŁUMIENIA ZAKŁÓCENIA ΔM_p) DLA WYBRANYCH REGULATORÓW WYZNACZONYCH METODAMI: LQG, PLACE-LQRY, KALMAN-PLACE, PLACE-PLACE ORAZ PD WYZNACZONYM DLA MODELU RZĘDU 21

Dla każdego analizowanego przypadku, regulatory wyznaczone metodami LQG oraz PLACE-LQRY, cechowała zdecydowanie wyższa zdolność tłumienia zakłócenia ΔM_p w porównaniu z regulatorem PD. Na rysunku 127 przedstawiono odpowiedź skokową układu zamkniętego z regulatorami LQG, PLACE-LQRY, oznaczonymi kolorem żółtym w tabeli 10, oraz regulatorem PD wyznaczonym z zastosowaniem modelu piętnastego rzędu.



RYSUNEK 127. CHARAKTERYSTYKI SKOKOWE UKŁADU ZAMKNIĘTEGO DLA ZAKŁÓCENIA ΔM_p

8.5. Uwagi

Modele parownika kotła energetycznego BP-1150, wykorzystywane w procesie projektowania układu sterowania (dla szeregu metod opisanych w rozdziale 8), uzyskano w drodze wieloetapowej dekompozycji i redukcji modelu pierwotnego o rzędzie około 15000.

Rozmiar modeli zredukowanych parownika użytych w celu projektowania układów regulacji, liczony razem z modelem układu przygotowania paliwa, wynosi od 10 do 33. Użycie modeli wyższego rzędu powoduje wystąpienie problemów numerycznych dla wykorzystywanych poleceń środowiska MATLAB.

Najistotniejsze zakłócenie, z punktu widzenia projektowania układu sterowania, dla parownika stanowi zmiana poboru pary z separatora ΔM_p . Widmo zakłóceń przepływem pary sięga częstotliwości $\omega_{max}=10^2 \text{ rad/s}$, przez co układy sterowania powinny cechować się możliwie wysokim tłumieniem zakłócenia do podanej częstotliwości. Dla toru regulacji ciśnienia $\Delta U_{UPP} \rightarrow \Delta P_{SEP}$ występuje bardzo duże opóźnienie, które wynika z obecności układu przygotowania paliwa (opóźnienie transportowe 50 s, stała czasowa 300 s). Dodatkowo model parownika stanowi obiekt niestabilny.

W podrozdziale 8.4 zawarto opis projektu układów sterowania metodami LQG, PLACE-LQRY, KALMAN-PLACE, PLACE-PLACE oraz regulatorów PD na podstawie zredukowanych modeli parownika. Wyznaczone układy regulacji zostały zastosowane do modelu wysokiego rzędu w celu wyznaczenia ich właściwości. Najlepszymi cechami jakościowymi charakteryzują się układy regulacji wyznaczone metodami LQG i PLACE-LQRY.

Układ regulacji wyznaczony na podstawie modelu siedemnastego rzędu dla metody LQG cechuje się wysokim poziomem tłumienia zakłóceń dla torów $\Delta M_p \rightarrow \Delta P_{SEP}$ (średnia wartość modułu transmitancji dla częstotliwości z przedziału $\langle 10^{-4} - 10^{-2} \text{ rad/s} \rangle$ wynosi 0,12) oraz $\Delta M_p \rightarrow \Delta H_{SEP}$ (średnia wartość modułu transmitancji dla częstotliwości z przedziału $\langle 10^{-4} - 10^{-2} \text{ rad/s} \rangle$ wynosi 0,02). Zapas stabilności dla toru regulacji ciśnienia wynosi odpowiednio 6 dB dla amplitudy oraz 39 stopni dla fazy.

Układ regulacji wyznaczony na podstawie modelu dziesiątego rzędu dla metody PLACE-LQRY posiada nieco gorsze właściwości dla tłumienia zakłócenia ΔM_p (średnia wartość modułu transmitancji dla toru $\Delta M_p \rightarrow \Delta P_{SEP}$ i częstotliwości z przedziału $\langle 10^{-4} - 10^{-2} \text{ rad/s} \rangle$ wynosi 0,15, dla toru $\Delta M_p \rightarrow \Delta H_{SEP}$ 0,03). Zapas stabilności dla toru regulacji ciśnienia wynosi odpowiednio 9 dB dla amplitudy oraz powyżej 40 stopni dla fazy.

Weryfikacja modeli zredukowanych parownika kotła BP-1150 byłaby w pełni możliwa, gdyby regulatory zaprojektowane na podstawie modeli zredukowanych, zastosować w realnym układzie sterowania, a następnie sprawdzić jakość układów regulacji. Niestety w przypadku takiego obiektu sterowania jest to niemożliwe. Uzyskane wyniki mogą stanowić podstawę do projektowania przyszłych układów sterowania.

Autor przeprowadził symulacyjną weryfikację modeli zredukowanych poprzez zastosowanie modeli zredukowanych niskiego rzędu do modelu parownika wysokiego rzędu. Uzyskane wyniki wskazują, że zaprojektowany układ sterowania zapewnia:

- znacznie (kilkukrotnie) silniejsze tłumienie zakłócenia ΔM_p niż klasyczny układ regulacji PD,
- znacznie krótsze przebiegi przejściowe po wystąpieniu zakłócenia.

Z przeprowadzonych analiz wynika, że do prawidłowego zaprojektowania układu sterowania wystarcza nawet model parownika czwartego rzędu (wraz z układem przygotowania paliwa – dziesiątego rzędu).

Podsumowanie

Redukcja rzędu modeli z zastosowaniem SVD charakteryzuje się złożonością obliczeniową $O(n^3)$, co pociąga za sobą bardzo długie czasy obliczeń. Ponadto wymogi co do rozmiaru pamięci operacyjnej dla modeli wysokiego rzędu przekraczają możliwości przeciętnych komputerów (np. przechowywanie macierzy kwadratowej wielkości 10^4 w formacie podwójnej precyzji, przy założeniu macierzy gęstej, wymaga około 1GB pamięci RAM). Ze względu na wysoki rząd rozważanego modelu parownika kotła BP-1500 (około 15000 zmiennych stanu) bezpośrednia redukcja rzędu modelu jest zadaniem praktycznie niewykonalnym z wykorzystaniem posiadanych narzędzi. Z powyższych powodów realizacja celu pracy związana jest z koniecznością użycia metod dekompozycyjnych modeli. Ze względu na konieczność zapewnienia adekwatności modeli zredukowanych w zadanym przedziale częstotliwości, do redukcji rzędu modeli zastosowano metodę częstotliwościowych funkcji wagowych. Metoda FW z zastosowaniem ewolucyjnego algorytmu doboru parametrów filtrów (przybornik eAMOR) wymaga realizacji wielu setek pojedynczych operacji redukcji w celu wyznaczenia optymalnych parametrów dla filtrów.

Rozdziały trzeci oraz czwarty zawierają główne rezultaty teoretyczne pracy, związane z dekompozycją modeli w celu niezależnej redukcji modeli zdekomponowanych.

W rozdziale 3 przedstawiono szereg metod dekompozycyjnych cechujących się różnymi właściwościami. Dekompozycja strukturalna, realizująca podział modelu na części o rzędzie stanowiącym wielokrotność rozmiaru elementu skończonego, może zostać zastosowana w przypadku gdy wydzielone podsystemy łączy rzadki interfejs (wydzielone podsystemy są powiązane niewielką liczbą zmiennych stanu). Dekompozycja strukturalna uproszczona, poprzez pominięcie interfejsów między podsystemami w procesie redukcji, jest celowa wyłącznie w przypadku podsystemów słabo powiązanych. Zaproponowana przez autora modyfikacja algorytmu dekompozycji Epsilon, powiązanej z dekompozycją slow-fast, w odniesieniu do modeli podsystemów parownika kotła energetycznego BP-1150, pozwoliła na opracowanie wielu nowych metod dekompozycji slow-fast: *Esf-Str*, *Esf-StrUp*, *Esf-SchSylv*, *Esf-SchComp*, *Esf-Sch*, *Esf-MLS*, *Esf-SchCompZm*. Opracowane metody w zdecydowanej większości cechują się niskim błędem modeli zredukowanych oraz wysokim odsetkiem modeli stabilnych. Dla analizowanych modeli zastosowanie jednej z metod *Esf* prowadzi do uzyskania podsystemów znacznie różniących się rozmiarem. Dekompozycja MLS przekształca modele podsystemów w taki sposób, aby pominięcie interfejsu nie powodowało pojawienia się błędu dla stanu ustalonego. W zaproponowanych przez autora metodach Schur Complement oraz zmodyfikowanej Schur Complement uwzględniany jest interfejs między podsystemami uzyskanymi metodą MLS podczas operacji scalania. Dekompozycja Schur'a umożliwia wydzielenie podsystemów bez sprzężenia zwrotnego, natomiast dekompozycja Schur-Sylvester wydziela dwa niezależne podsystemy połączone równolegle.

Wadą stosowanej metodologii redukcji modeli wysokiego rzędu jest możliwość uzyskania niestabilnego modelu zredukowanego, uzyskanego na podstawie scalenia zdekomponowanych modeli zredukowanych. Problem niestabilności pojawia się w sytuacji, gdy między zdekomponowanymi podsystemami funkcjonują silne sprzężenia zwrotne (szczególnie silnie oddziałujące na wysokich częstotliwościach). Na skutek modyfikacji modeli podsystemów w wyniku ich redukcji, rezultatem scalenia może okazać się model niestabilny. Z przeprowadzonych obliczeń wynika, że w celu uzyskania stabilnych modeli zredukowanych, niezbędna jest aproksymacja charakterystyk częstotliwościowych w szerokim zakresie częstotliwości (w przypadku modeli parownika nawet do 1000 rad/s). Pociąga to za sobą wysokie rzędy zredukowanych modeli podsystemów. W literaturze można znaleźć prace, w których analizowany jest problem stabilności modeli zredukowanych, jednak

konieczne są dalsze prace pozwalające na określenie precyzyjnych warunków, jakie należy spełnić podczas redukcji modeli podsystemów, aby model scalony był stabilny [155] [156]. Jest to problem, którym autor chciałby się zająć w dalszych pracach.

Przedstawione metody dekompozycji jednopoziomowej posłużyły do opracowania metod dekompozycji dwupoziomowych (rozdział 4). Założono możliwość realizacji odmiennych metod dekompozycji na pierwszym i drugim poziomie, w związku z czym całkowita liczba przeanalizowanych w pracy metod dekompozycji dwupoziomowych wynosi 78. Pierwszy poziom dekompozycji może zostać zrealizowany z wykorzystaniem jednej z trzynastu metod opisanych w rozdziale 3. Drugi poziom dekompozycji realizowany jest z pominięciem dekompozycji slow-fast. Dla analizowanych modeli, wydzielony podsystem szybki posiada niewielki rozmiar, przez co realizacja dekompozycji slow-fast na poziomie drugim jest niecelowa, przede wszystkim ze względu na czas obliczeń.

Rozdział 6 zawiera krótki opis modeli matematycznych parownika kotła BP-1150 i został przedstawiony na podstawie prac [32] [15]. Główne rezultaty praktyczne pracy zawarte są w rozdziale siódmym, w którym przedstawiono wyniki redukcji modeli podsystemów parownika. Opracowane metody dekompozycji jednopoziomowej oraz dwupoziomowej zostały poddane weryfikacji z zastosowaniem modeli niskiego rzędu. Dla każdej metody dekompozycji wyznaczono 306 modeli zredukowanych zmieniając zakres aproksymacji charakterystyk częstotliwościowych dla metody FW oraz rząd modelu zredukowanego. Przy ocenie jakościowej danej metody dekompozycji posłużono się miarą błędu aproksymacji charakterystyk częstotliwościowych dla zadanego przedziału częstotliwości, odsetkiem zredukowanych modeli stabilnych oraz orientacyjnym czasem redukcji. W wyniku przeprowadzonych analiz, do redukcji modeli podsystemów parownika wysokiego rzędu wybrano następujące dekompozycje dwupoziomowe: *Str/Str*, *Str/SchSylv*, *Sch/SchSylv*, *Esf-SchSylv/SchSylv*, *Esf-SchSylv/SchComp*, *Esf-Sch/SchSylv*.

Na podstawie przeprowadzonych operacji redukcji modeli podsystemów rur ekranowych można przedstawić następujące wnioski:

- Dekompozycja *Str/Str* cechuje się najmniejszym błędem redukcji (około 10^{-10} dla błędu średniokwadratowego względnego w przedziale częstotliwości od 10^{-3} do 20 rad/s). Aproksymacja charakterystyk częstotliwościowych modelu oryginalnego w szerokim zakresie częstotliwości (od 10^{-3} do 10^3) zapewnia uzyskanie wysokiego odsetku modeli stabilnych.
- Akceptowalną metodą dekompozycji dla rozważanych modeli jest dekompozycja *Str/SchSylv*. Cechuje się ona niskim błędem aproksymacji dla modeli zredukowanych wyższego rzędu, przy wysokim odsetku modeli stabilnych.
- Dekompozycja *Sch/SchSylv* pod względem błędu aproksymacji modeli zredukowanych wypada najgorzej. Spowodowane jest to silnym wzmocnieniem interfejsów wydzielonych podsystemów, co przy niewielkich błędach aproksymacji podsystemów zredukowanych powoduje wysoki błąd modelu scalonego. Metoda dla analizowanych modeli zredukowanych zapewniała stu procentowy odsetek modeli stabilnych.
- Dekompozycje slow-fast dają zawsze stabilne modele zredukowane, których błąd aproksymacji w zdecydowanej większości przypadków można sprowadzić (poprzez zwiększenie rozmiaru zredukowanej części szybkiej) do wartości porównywalnej dla dekompozycji *Str/Str*. Mając na uwadze większą złożoność obliczeniową redukcji dla dekompozycji slow-fast (zdecydowanie dłuższe czasy obliczeń) oraz niski błąd aproksymacji modeli zredukowanych, celowe wydaje się zastosowanie dekompozycji trypoziomowej, gdzie pierwszy poziom stanowiłaby dekompozycja slow-fast, a dwa

kolejne poziomy dotyczyłyby części wolnej. Złożoność obliczeniowa redukcji zostałaby wtedy sprowadzona do poziomu porównywalnego z dekompozycjami nie należącymi do grupy slow-fast.

- Podobny wniosek nasuwa się w stosunku do dekompozycji Str/Str – niski błąd aproksymacji modeli zredukowanych pozwala przypuszczać, że dodanie kolejnego poziomu dekompozycji pozwoliłoby na otrzymanie modeli zredukowanych o niskim błędzie aproksymacji w zdecydowanie krótszym czasie. Problem, który może się pojawić w przypadku dekompozycji trzypoziomowej może natomiast dotyczyć odsetku stabilnych modeli wynikowych. Wspomniane przypuszczenia staną się przedmiotem przyszłych prac autora.

Przeprowadzenie tak dużej liczby eksperymentów:

- analiza 13 metod dekompozycji jednopoziomowej:

$$\begin{array}{cccccc}
 13 & * & 306 & * & 2 & * & (25+10) & = & 278460 \\
 \text{metod} & & \text{przypadków} & & \text{Modele} & & \text{redukcji} & & \text{redukcji SVD} \\
 & & & & \text{dekomponowane} & & \text{metoda FW} & &
 \end{array}$$

- analiza 78 metod dekompozycji dwupoziomowej (z czego 52 metody dawały poprawne modele) :

$$\begin{array}{cccccc}
 26 & * & 306 & * & 4 & * & (25+10) & = & 1113840 \\
 \text{metod} & & \text{przypadków} & & \text{Modele} & & \text{redukcji} & & \text{redukcji} \\
 (\text{poza } Esf) & & & & \text{zdekomponowane} & & \text{metoda FW} & & \text{SVD} \\
 26 & * & 306 & * & 3 & * & (25+10) & = & 835380 \\
 \text{metod} & & \text{przypadków} & & \text{Modele} & & \text{redukcji} & & \text{redukcji} \\
 (Esf) & & & & \text{zdekomponowane} & & \text{metoda FW} & & \text{SVD}
 \end{array}$$

- analiza 6 metod dekompozycji dwupoziomowej dla czterech modeli podsystemów rur ekranowych rzędu 3300:

$$\begin{array}{cccccc}
 4 & * & 3 & * & 4 & * & (500+300+150) & = & 45600 \\
 \text{Modele} & & \text{metody} & & \text{modele} & & \text{redukcji} & & \text{redukcji} \\
 & & (\text{poza } Esf) & & \text{dekomponowane} & & \text{metoda FW} & & \text{SVD} \\
 4 & * & 3 & * & 3 & * & (500+300+150) & = & 34200 \\
 \text{Modele} & & \text{metody} & & \text{modele} & & \text{redukcji} & & \text{redukcji} \\
 & & (Esf) & & \text{dekomponowane} & & \text{metoda FW} & & \text{SVD}
 \end{array}$$

bez wykorzystania środowiska równoległego byłoby wysoce problematyczne (ze względu na długi czas obliczeń). Aby zrealizować cel pracy należało stworzyć klaster obliczeniowy oraz programy wykonujące równoległą dekompozycję i redukcję analizowanych modeli. Zaprojektowany klaster komputerowy stanowi system heterogeniczny, składa się z 8 komputerów wyposażonych w procesory wielordzeniowe oraz procesory graficzne. Opracowane programy, przeznaczone do realizacji obliczeń równoległych w środowisku MATLAB oraz z wykorzystaniem procesorów graficznych realizują setki algorytmów związanych między innymi z rozwiązywaniem równania Lapunowa, wyznaczaniem macierzy transformacji redukcji T i L, wykonywaniem dekompozycji jedno i wielopoziomowych, skalaniem zredukowanych modeli, a także organizacją obliczeń równoległych. W celu doboru optymalnych parametrów filtrów dla metody redukcji FW wykorzystano rezultaty pracy doktorskiej dr inż. Marka Rydla [15], modyfikując opracowany przez niego przybornik eAMOR dla potrzeb redukcji równoległej.

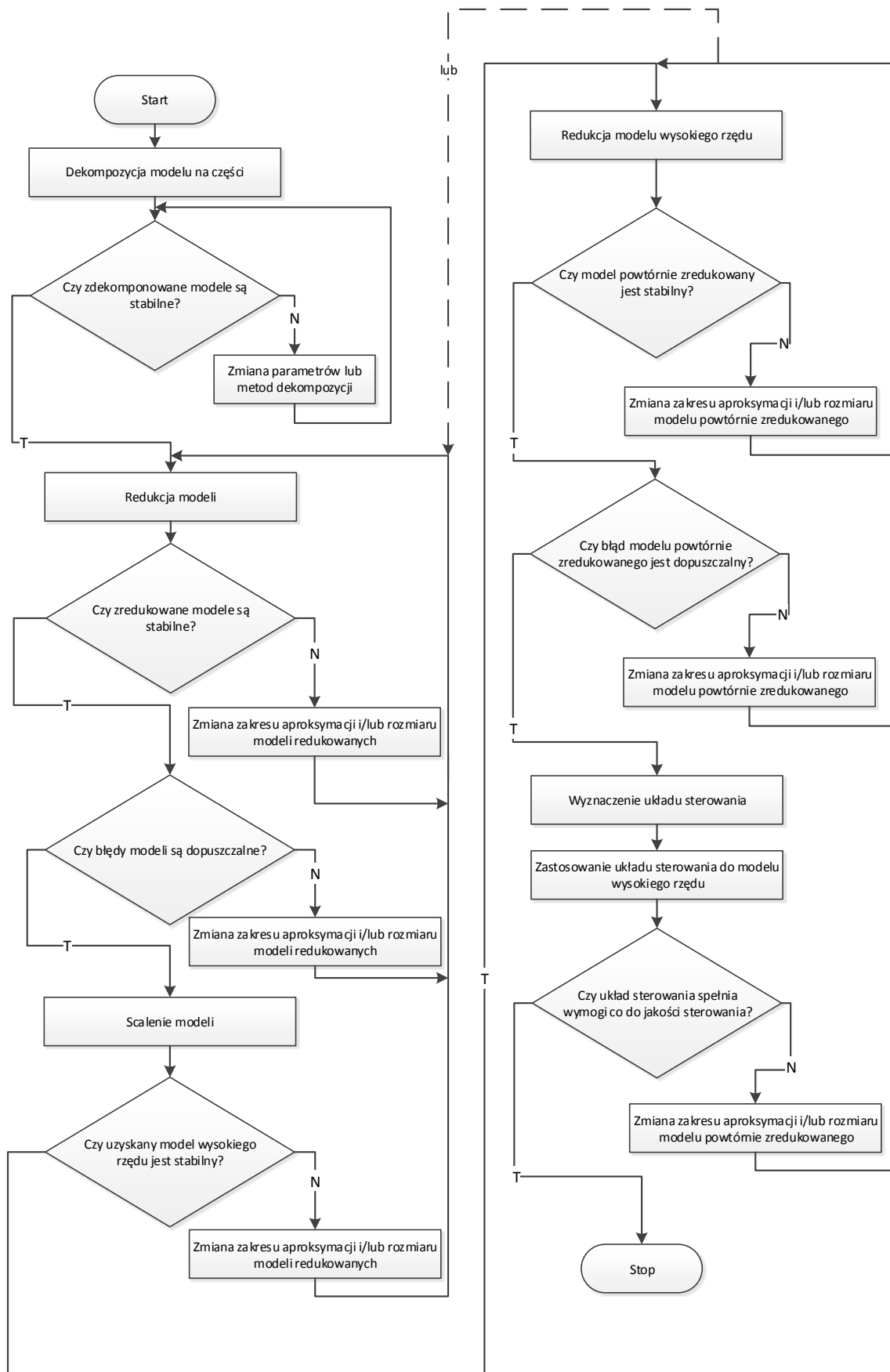
Modyfikacja algorytmu dekompozycji Epsilon, opracowanego przez Dragoslava Šiljaka [30], pozwoliła na opracowanie metod i programów dla dekompozycji slow-fast. Dekompozycja MLS przedstawiona w pracach Franka Blömelinga [31] [123] stanowiła podstawę dla opracowania metod i programów dekompozycji Schur-Complement i zmodyfikowanej Schur-Complement.

Dorobek praktyczny rozprawy obejmuje implementację metod dekompozycji jedno i wielopoziomowych a także opracowanie programów realizujących redukcję równoległą

z wykorzystaniem środowiska klastrowego MATLAB i procesorów graficznych metodą częstotliwościowych funkcji wagowych.

W celu weryfikacji uzyskanych modeli zredukowanych parownika kotła BP-1150, w rozdziale 8 zaprojektowano układ sterowania dla modelu wysokiego rzędu parownika kotła energetycznego BP-1150 z zastosowaniem modeli zredukowanych. Ze względu na brak możliwości analizy zaprojektowanych układów regulacji w odniesieniu do modelu pierwotnego (około 15000 zmiennych stanu), za model odniesienia przyjęto model wstępnie zredukowany. Dla zredukowanych modeli parownika, z dołączonym modelem układu przygotowania paliwa, których rząd wynosił od 10 do 33, wyznaczono układy sterowania z wykorzystaniem metod LQG, PLACE-LQRY, KALMAN-PLACE, PLACE-PLACE. Uzyskane układy regulacji porównano z klasycznym układem regulacji, zawierającym regulatory PD. Przyjęto kryterium jakościowe oceny otrzymanych układów sterowania w postaci zapasu stabilności dla toru regulacji ciśnienia oraz tłumienia głównego zakłócenia – zmiany poboru pary z separatora ΔM_p . Dla metod LQG oraz PLACE-LQRY otrzymano jakościowo najlepsze układy regulacji, które cechują się wysokim zapasem stabilności i wysokim stopniem tłumieniem zakłócenia ΔM_p . Zaprojektowany układ regulacji LQG cechuje się znacznie lepszymi właściwościami niż klasyczny układ regulacji, co potwierdza celowość zastosowanego w rozprawie podejścia do modelowania modeli obiektów sterowania o złożonej strukturze.

Na podstawie zdobytych doświadczeń podczas realizacji rozprawy, można zaproponować następującą metodologię wyznaczania modelu obiektu sterowania o złożonej strukturze z wykorzystaniem wielopoziomowej dekompozycji modelu, redukcji modeli podsystemów oraz scaleniem dekomponowanych modeli zredukowanych. Schemat blokowy realizowanych działań, prowadzących do otrzymania zadawalającego jakościowo modelu zamieszczono na rysunku 128.



RYSUNEK 128. SCHEMAT BLOKOWY REALIZOWANYCH DZIAŁAŃ, PROWADZĄCYCH DO OTRZYMANIA ZADAWAJĄCEGO JAKOŚCIOWO UKŁADU STEROWANIA

Literatura

1. **Stolarski T., Nakasone S. i Yoshimoto S.** *Engineering analysis with ANSYS software*, Oxford, Elsevier Butterworth-Heinemann, 2006.
2. **ABAQUS Inc.** *Scripting User's Manual*, United States of America, ABAQUS Inc., 2004.
3. **MSC Software Corporation.** *MSC Nastran 2007 r1 Implicit Nonlinear (SOL 600) User's Guide*, U.S.A., MSC Software Corporation, 2007.
4. **Kurowski P.** *Engineering Analysis with COSMOSWorks Professional, Finite Element Analysis with COSMOSWork 2005*, Madison, Schroff Development Corporation, 2005.
5. **Logan D.** *A first course in the finite element method using algor*, Boston, PWS Pub., 1997.
6. **MSC Software.** *Patran 2008 r1 Release Guide*, U.S.A., MSC Software, 2008.
7. **Altair Engineering Inc.** *Altair HyperWorks 8.0, HyperMesh 8.0 User's Guide*, Altair Engineering Inc., 2007.
8. **Cook D., Malkus D. i Plesha M.** *Concepts and Applications of Finite Element Analysis, 4th Edition*, United States, Wiley and Sons, 2001.
9. **Rudnyi E. i Korvink J.** *Model Order Reduction For Large Scale Finite Element Engineering Models*, European Conference on Computational Fluid Dynamics ECCOMAS CDF 2006, 2006.
10. **Antoulas A.** *Approximation of Large-Scale Systems*, Philadelphia, SIAM, 2005.
11. **Schelfhout G.** *Model reduction for control design*, Heverlee, Katholieke Universiteit Leuven, 1996.
12. **Benner P. i inni.** *State-Space Truncation Methods for Parallel Model Reduction of Large-Scale Systems*, *Parallel Comput.*, 29, pp. 1701-1722, 2003.
13. **Varga A. i Anderson B.** *Accuracy enhancing methods for the frequency-weighted balancing related model reduction*, *Proc. of CDC'2001*, Orlando, Florida, pp. 3659-3664, 2001.
14. **Rydel M.** *eAMOR evolutionary Algorithms for Model Order Reduction Users's Guide*, Opole, 2011.
15. —. *Zredukowane hierarchiczne modele złożonych obiektów sterowania na przykładzie kotła energetycznego*, Opole, Wydział Elektrotechniki, Automatyki i Informatyki Politechniki Opolskiej, Rozprawa Doktorska, 2008.
16. **Stanisławski W. i Raczyński D.** *Rozdział: Metody wyznaczania gramianów sterowalności i obserwowalności dla ciągłych stacjonarnych układów liniowych*, *W: Postępy Automatyki i Robotyki, KAIR PAN*, Kielce, Wydawnictwo Politechniki Świętokrzyskiej Monografie Tom 16, pp. 571-580, 2011.
17. **Dębowski A.** *Automatyka podstawy teorii*, Warszawa, Wydawnictwa Naukowo-Techniczne, 2008.
18. **Baur U. i Benner P.** *Gramian-Based Model Reduction for Data-Sparse Systems*, Chemnitz, Chemnitz Scientific Computing Preprints, 2007.
19. **Sanchez-Gasca J. i Chow J.** *Power system reduction to simplify the design of damping controllers for interarea oscillations*, *IEEE Trans. Power Syst.*, vol. 11, no. 3, pp. 1342–1349, 1996.
20. **Glover K.** *All optimal Hankel-norm approximations of linear multivariable systems and their L_∞ - error bounds*, *Int. J. Control*, vol. 39, pp. 1115–1193, 1984.
21. **Liu Y. i Anderson B.** *Model reduction with time delay*, *IEE Proceedings - Control Theory and Applications*, vol.134, no.6, pp. 349-367, 1987.
22. **Lin C. i Chiu T.** *Model reduction via frequency weighted balanced realization*, *Control Theory and Advanced Technol.*, vol. 8, pp. 341-451, 1992.
23. **Gawronski W. i Juang J.** *Model reduction in limited time and frequency intervals*, *Int. J. System Sci.*, vol. 21, no. 2, pp. 349-376, 1990.

24. **Gugercin S. i Antoulas A.** *A time-limited balanced reduction method*, Proc. of the 42nd IEEE Conference on Decision and Control., vol. 5, pp. 5250-5253, 2003.
25. **Varga A.** *Numerical approach for frequency-weighted Hankel-norm approximation*, Proc. of European Control Conf., pp. 640-645, 2001.
26. **Stanisławski W. i Raczyński D.** *Programowanie systemowe mikroprocesorów rodziny x86*, Warszawa, Wydawnictwo Naukowe PWN, 2010.
27. —. *Programowanie jednostek wektorowych procesorów rodziny x86*, Opole, Oficyna Wydawnicza Politechniki Opolskiej, 2009.
28. **Raczyński D.** *Vector calculations using x86 processors family*, Opole-Pokrzywna, Zeszyty Naukowe Politechniki Opolskiej, seria Elektryka, z. 63, pp. 49-50, 2010.
29. **Kirk D. i Hwu W.** *Programming Massively Parallel Processors: A Hands-on Approach*. San Francisco, Morgan Kaufmann, 2010.
30. **Šiljak D.** *Decentralized Control of Complex Systems*, New York, Academic Press, 1991.
31. **Blömeling F.** *Multi-level substructuring combined with model order reduction methods*, Linear Algebra and its Applications, vol. 436, pp. 3864-3882, 2012.
32. **Stanisławski W.** *Modelowanie i symulacja komputerowa parowników przepływowych kotłów energetycznych*, *Studia i Monografie*, z. 124, Opole, Oficyna Wydawnicza Politechniki Opolskiej, 2001.
33. **Sootla A.** *Model Reduction Using Semidefinite Programming*, Lund, Department of Automatic Control, Lund University, 2009.
34. **Sabino J.** *Solution of Large-Scale Lyapunov Equations via the Block Modified Smith Method*, Houston, Rice University, phd thesis, 2006.
35. **Raczyński D.** *Comparison of methods for determining controllability and observability gramians for linear time invariant systems*, Opole-Pokrzywna, Zeszyty Naukowe Politechniki Opolskiej, seria Elektryka, z. 66, pp. 77-78, 2011.
36. **Li J.-R.** *Model Reduction of Large Linear Systems via Low Rank System Gramians*. Massachusetts, Massachusetts Institute Of Technology, phd. Thesis, 2000.
37. **Petkov P. i Christov N.** *Solving Lyapunov and Sylvester Matrix Equations with Condition and Accuracy Estimates*, Sofia, Department of Automatics, Technical University of Sofia, 1996.
38. **Al-Baiyat S. i Bettayeb M.** *A new model reduction scheme for K-POWER bilinear systems*, San Antonio, Proceedings of the 32nd Conference on Decision and Control, pp. 22-27, 1993.
39. **Hu Q. i Cheng D.** *The polynomial solution to the Sylvester matrix equation*, Applied Mathematics Letters 19(9), pp. 859-864, 2006.
40. **Gajić Z.** *Lyapunov Matrix Equation in System Stability and Control*, San Diego, Academic Press, 1995.
41. **Zhou Y.** *Numerical Methods for Large Scale Matrix Equations with Applications in LTI System Model Reduction*, Houston, Rice University, phd thesis, 2002.
42. **Stykel T.** *Numerical Solution and Perturbation Theory for Generalized Lyapunov Equation*, Linear Algebra and its Applications, 349(1-3), pp. 155-185, 2002.
43. **Lee Y.** *The iterative methods for computing invariant subspaces and their applications*, Sei, Department of Mathematics and Statistics, Sei University, Ms thesis, 1987.
44. **Granat R.** *Algorithms and Library Software for Periodic and Parallel Eigenvalue Reordering and Sylvester-Type Matrix Equations with Condition Estimation*, UMEÅ, Department of Computing Science, UMEÅ UNIVERSITY, phd. thesis, 2007.
45. **Gloub G., Nash S. i Van Loan C.** *A Hessenberg-Schur Method for the Problem $AX+XB=C$* , IEEE Transactions on Automatic Control, vol. AC-24, No. 6, pp. 909-913, 1979.
46. **Smith R.** *Matrix equation $XA+BX=C$* , SIAM J. Appl. Math., vol. 16, pp. 198-201, 1968.

47. **Benner P., Quintana-Ortí E. i Quintana-Ortí G.** *Solving Stable Stein Equations on Distributed Memory Computers*, Euro-Par'99 Proceedings of 5th International Euro-Par Conference on Parallel Processing, pp. 1120-1123, 1999.
48. **Anderson B.** *Second-order convergent algorithms for the steady-state Riccati equation*, Int. J. Control, vol. 28, pp. 295-306, 1978.
49. **Penzl T.** *A cyclic low rank Smith method for large sparse Lyapunov equations with applications in model reduction and optimal control*, SIAM J. SCI. COMPUT., vol. 21, no. 4, pp. 1401–1418, 2000.
50. **Wong N. i Balakrishnan V.** *Quadratic Alternating Direction Implicit Iteration for the Fast Solution of Algebraic Riccati Equations*. Hong Kong, In Proc. 2005 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), 2005.
51. **Benner P., Li R. i Truhar N.** *On the ADI method for Sylvester equations*, Journal of Computational and Applied Mathematics, vol. 233, Issue 4, pp. 1035-1045, 2009.
52. **Wachspress E.** *The ADI minimax problem for complete spectra*, Appl. Math. Lett., Vol. 1, pp. 311-314, 1988.
53. **Kenney C. i Laub A.** *The matrix sign function*, IEEE Trans. Automat. Control, vol. 40, no. 8, pp. 1330–1348, 1995.
54. **Benner P.** *Contributions to the Numerical Solution of Algebraic Riccati Equations and Related Eigenvalue Problems*, Berlin, Logos Verlag, 1997.
55. **Roberts J.** *Linear model reduction and solution of the algebraic Riccati equation by use of the sign function*, Internat. J. Control, 32(4), pp. 677–687, 1980.
56. **Byers R.** *Solving the algebraic Riccati equation with the matrix sign function*, Linear Algebra Appl., 85, pp. 267–279, 1987.
57. **Gardiner J. i Laub A.** *A generalization of the matrix-sign-function solution for algebraic Riccati equations*, Internat. J. Control, 44(3), pp. 823–832, 1986.
58. **Lancaster P. i Rodman L.** *The Algebraic Riccati Equation*, Oxford, Oxford University Press, 1995.
59. **Li J.-R. i White J.** *Low rank solution of Lyapunov equation*, SIAM J. Matrix Anal. Appl., vol. 24, no. 1, pp. 260-280, 2002.
60. **Hammarling S.** *Numerical Solution of the Stable, Non-negative Definite Lyapunov Equation*, IMA Journal of Numerical Analysis, vol. 2, pp. 303-323, 1982.
61. **Benner P. i Quintana-Ortí E.** *Solving stable generalized Lyapunov equations with the matrix sign function*, Numer. Algorithms, 20(1), pp. 75-100, 1999.
62. **Larin V. i Aliev F.** *Construction of square root factor for solution of the Lyapunov matrix equation*, Sys. Control Lett., vol. 20, Issue 2, pp. 109-112, 1993.
63. **Benner P., Quintana-Ortí E. i Quintana-Ortí G.** *Model Reduction of Large-Scale Dense Systems*, Proceedings of the IEE UKACC International Conference CONTROL 2000, Cambridge, UK, 2000.
64. **Quintana-Ortí E. i Geijn R.** *Specialized Parallel Algorithms for Solving Linear Matrix Equations in Control Theory*, Journal of Parallel and Distributed Computing, 61, pp. 1489-1504, 2001.
65. **Baur U.** *Low Rank Solution of Data-Sparse Sylvester Equations*, Numer. Linear Alg. Appl., 15 (2008), pp. 837–851, 2008.
66. **Benner P., Mehrmann V. i Sorensen D. (edytorzy).** *Dimension Reduction of Large Scale Systems*. Berlin/Heidelberg, vol. 45 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, 2005.
67. **Benner P., Claver J. i Quintana-Ortí E.** *Efficient solution of coupled Lyapunov equations via matrix sign function iteration*, In: A. Dourado et al. (ed.) Proc. 3rd Portuguese Conf. on Automatic Control CONTROL'98, Coimbra, pp. 205-210, 1998.

68. **Benner P. i inni.** *Accelerating Model Reduction of Large Linear Systems with Graphics Processors*, Applied Parallel and Scientific Computing, Lecture Notes in Computer Science Volume 7134, pp. 88-97, 2012.
69. **Gugercin S., Sorensen C. D. i Antoulas A.** *A modified low-rank Smith method for large-scale Lyapunov equations*, Numerical Analysis, Vol. 32, Issue 1, pp. 27-55, 2003.
70. **Raczyński D.** *Cholesky decomposition of controllability and observability gramians using parallel computation with GPU*, Opole-Pokrzywna, Zeszyty Naukowe Politechniki Opolskiej, seria Elektryka, z. 68, pp. 59-60, 2012.
71. **Benner P. i Saak J.** *A Galerkin-Newton-ADI Method for Solving Large-Scale Algebraic Riccati Equations*, DFG Priority Programme Optimization with Partial Differential Equations, Preprint SPP1253-090, 2010.
72. **Saak J.** *Efficient Numerical Solution of Large Scale Algebraic Matrix Equations in PDE Control and Model Order Reduction*, Chemitz, Faculty of Mathematics, Chemnitz University of Technology, Dissertation, 2009.
73. **Nowakowski C. i inni.** *Model Reduction of an Elastic Crankshaft for Elastic Multibody Simulations*, ZAMM - Journal of Applied Mathematics and Mechanics, 2012.
74. **Mitkowski W.** *RÓWNANIA MACIERZOWE i ich zastosowanie*. Kraków, Uczelniane Wydawnictwa Naukowo-Dydaktyczne AGH, 2006.
75. **Power H.** *A note on matrix equation $ATLA-L=-K$* , IEEE trans. Automatic Control, vol 14, pp. 411-412, 1969.
76. **Raczyński D. i Stanisławski W.** *Controllability and observability gramians parallel computation using GPU*, Gdańsk, Journal of theoretical and applied computer science, vol. 6, no. 1, pp. 47-66, 2012.
77. **Peaceman D. i Rachford H.** *The numerical solutions of parabolic and elliptic differential equations*, J. SIAM, 3, pp. 28-41, 1955.
78. **Gugercin S. i Li J.-R.** *Smith type methods for balanced truncation of large sparse systems*, Oberwolfach, Proceedings of the Oberwolfach mini-workshop on dimension reduction, Lecture Notes in Computational Science and Engineering, 2004.
79. **Penzl T.** *Eigenvalue decay bounds for solutions of Lyapunov equations: The symmetric case*, Systems Control Lett., 40, pp. 139-144, 2000.
80. **Vemuri B. i Lai S.** *A Fast Solution to the Surface Reconstruction Problem*, SPIE, Vol. 2031 Geometric Methods in Computer Vision II (1993), pp. 27 - 37, 1993.
81. **Penzl T.** *LYAPACK: A MATLAB toolbox for large Lyapunov and Riccati equations, model reduction problems, and linear-quadratic optimal control problems, users' guide (ver. 1.0)*, 2000.
82. **Higham N.** *Functions of Matrices. Theory and Computation*. Philadelphia, SIAM, 2008.
83. —. *Newton's method for the matrix square root*, Math. Comp., 46, pp. 537-549, 1986.
84. **Sima V. i Benner P.** *Experimental evaluation of new SLICOT solvers for linear matrix equations based on the matrix sign function*, In Proceedings of 2008 IEEE Multi-conference on Systems and Control. 9th IEEE International Symposium on Computer-Aided Control Systems Design (CACSD), pp. 601-606, 2008.
85. **Bischof C. i Quintana-Ortí G.** *Computing rank-revealing QR factorizations of dense matrices*, ACM Trans. Math. Software, 24(2), pp. 226-253, 1998.
86. **Chan T.** *Rank revealing QR factorizations*, Linear Algebra Appl., 88/89, pp. 67-82, 1987.
87. **Miranian L. i Gu M.** *Strong rank revealing LU factorizations*, Linear Algebra Appl., 367, pp. 1-16, 2003.
88. **Pan C.-T.** *On the existence and computation of rank-revealing LU factorizations*, Linear Algebra Appl., 316, pp. 199-222, 2000.

89. **Moore B.** *Principal component analysis in linear systems: Controllability, observability, and model reduction*, IEEE Trans. Automat. Contr., vol. AC-26, pp. 17–32, 1981.
90. **Streif S., Findeisen R. i Bullinger E.** *Relating cross Gramians and sensitivity analysis in systems biology*, Kyoto, Proc. of the 17th International Symposium on Mathematical Theory of Networks and Systems, 24-28 July, Kyoto, Japan, pp. 437-442, 2006.
91. **Reynolds D.** *Coarse graining and control theory model reduction*, submitted to J. Stat. Phys., 2003.
92. **Philips J., Daniel L. i Silveira L.** *Guaranteed Passive Balancing Transformations for Model Order Reduction*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 22, No. 8, pp. 1027-1041, 2003.
93. **Lutowska A.** *Model Order Reduction for Coupled Systems using Low-rank Approximations*, Eindhoven, Technische Universiteit Eindhoven, phd. thesis, 2012.
94. **Raczyński D.** *A comparison of methods for calculation of transformation matrices for model order reduction using GPU parallel computing*, Opole-Pokrzywna, Zeszyty Naukowe Politechniki Opolskiej, seria Elektryka, z. 69, pp. 91-92, 2013.
95. **Fujimoto K. i Scherpen J.** *Balanced Realization and Model Order Reduction for Nonlinear Systems Based on Singular Value Analysis*, SIAM J. Control Optim., 48(7), pp. 4591–4623, 2010.
96. **Laub A. i inni.** *Computation of System Balancing Transformations and Other Applications of Simultaneous Diagonalization Algorithms*, IEEE Trans. on Automatic Control, vol. AC-32, no. 2, pp. 115-122, 1987.
97. **Varga A.** *Numerical methods and software tools for model reduction*, Proc. of 1st MATHMOD Conference, Wien, vol. 2, pp. 226-230, 1994.
98. —. *Balancing-free square-root algorithm for computing singular perturbation approximations*, Proc. of 30th IEEE CDC, vol. 2, pp. 1062-1065, 1991.
99. —. *Efficient Minimal Realization Procedure Based on Balancing*, Proc. of IMACS/IFAC Symposium on Modelling and Control of Technological Systems, Lille, France, vol. 2, pp. 42-47, 1991.
100. —. *Coprime Factors Model Reduction Based on Accuracy Enhancing Techniques*, Gordon and Breach Science Publishers S.A., SAMS, Vol. 11, pp. 303-311, 1993.
101. **Safonov M. i Chiang R.** *A Schur Method for Balanced-Truncation Model Reduction*, IEEE Trans. Automat. Contr., vol. 34, no. 7, 1989, pp. 729-733, 1989.
102. **Golub G. i Van Loan C.** *Matrix Computations. 3rd Edition*, Baltimore, The Johns Hopkins University Press, 1966.
103. **Meyer C.** *Matrix Analysis and Applied Linear Algebra*, Philadelphia, SIAM, 2000.
104. **Rowley C.** *Model reduction for fluids using balanced proper orthogonal decomposition*, International Journal of Bifurcation and Chaos, vol. 15, no. 3, pp. 997–1013, 2005.
105. **Obinata G. i Anderson B.** *Model Reduction for Control System Design*, London, Springer-Verlag, 2001.
106. **Driessel K.** *On Computing Eigenvalues Using a Toda Flow With Shifts*, Iowa, A Conference at Iowa State University, 2002.
107. **Enns D.** *Model reduction with balanced realizations: An error bound and a frequency weighted generalization*, Proc. 23-th IEEE Conf. Dec. Control, Las Vegas, pp. 127-132, 1984.
108. **Tsubakino D. i Fujimoto K.** *Weighted balanced realization and model reduction for nonlinear systems*, Transactions of the Society of Instrument and Control Engineers, Vol. 44, No. 1, pp. 44-51, 2008.
109. **Sreeram V. i Sahlan S.** *Improved Results on Frequency Weighted Balanced Truncation*, Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference, NJ, USA, CD-ROM, pp. 3250-3255, 2009.

110. **Schilders W., Vorst H. i Rommes J.** *Model Order Reduction. Theory, Research Aspects and Applications*, Berlin, Springer, 2008.
111. **Sreeram V. i Ghafoor A.** *Frequency Weighted Model Reduction Technique with Error Bounds*, IEEE American Control Conference, June 8-10, Portland, OR, USA, pp. 2584-2589, 2005.
112. **Li L. i Paganini F.** *Structured frequency weighted model reduction*, Decision and Control, 2003. Proceedings. 42nd IEEE Conference on, vol. 3, pp. 2841-2846, 2003.
113. **Wang G., Sreeram V. i Liu W.** *A New Frequency-Weighted Balanced Truncation Method and an Error Bound*, IEEE Transactions On Automatic Control, vol. 44, no. 9, pp. 1734-1737, 1999.
114. **Chiu T.** *Model Reduction by the Low-Frequency Approximation Balancing Method for Unstable Systems*, IEEE Trans. Automat. Contr., vol. 41, no. 7, pp. 995-997, 1996.
115. **Sou K. C.** *Convex Optimization Methods for Model Reduction*, Massachusetts, Phd Massachusetts Institute of Technology, 2008.
116. **Yang Y. i inni.** *Applied Numerical Methods Using MATLAB*, New Jersey, John Wiley & Sons, 2005.
117. **Heck B. i Haddad A.** *Singular Perturbation in Piecewise- Linear Systems*, IEEE Transactions on Automatic Control, vol. AC- 34, pp. 87-90, 1989.
118. **Naidu D. i Price D.** *Singular perturbations and time scale in the design of digital flight control systems*, NASA Technical Paper 2844, 1988.
119. **Kokotovic P., Khalil H. i O'Reilly J.** *Singular Perturbations Method in Control: Analysis and Design*, London, Academic Press, 1986.
120. **Sezer M. i Šiljak D.** *Nested Epsilon Decompositions of Linear Systems: Weakly Coupled and Overlapping Blocks*, SIAM. J. Matrix Anal. & Appl., 12(3), pp. 521–533, 1991.
121. **Zečević A. i Šiljak D.** *Control of Complex Systems. Structural Constraints and Uncertainty*, London, Springer, 2010.
122. **Sezer M. i Šiljak D.** *Nested ε decompositions and clustering of complex systems*, Automatica, 22, pp. 321–331, 1986.
123. **Blömeling F.** *Multi-level substructuring method for balanced-truncation model reduction*, Ph.D. thesis, Hamburg, Institute of Numerical Simulation. Hamburg University of Technology, 2008.
124. **Stewart G.** *Matrix Algorithms. Volume II: Eigensystems*, Philadelphia, SIAM, 2001.
125. **Pacheco P.** *An Indtroduction to Parallel Programming*, Burlington, Morgan Kaufmann, 2011.
126. **Wittwer T.** *An Introduction to Parallel Programming*, Leeghwaterstraat, VSSD, 2006.
127. **Owens J. i inni.** *GPU Computing*, Proceedings of the IEEE, Vol. 96, No. 5, pp. 879-899, 2008.
128. **Nickolls J. i Dally W.** *The GPU Computing Era*, IEEE Micro, vol. 30, no. 2, pp. 56-69, 2010.
129. **Lezar E.** *GPU Acceleration of Matrix-based Methods in Computational Electromagnetics*, phd. Thesis, Stellenbosch University, Department of Electrical and Electronic Engineering, 2011.
130. **Farber R.** *CUDA Application Design and Development*, San Francisco, Morgan Kaufmann, 2011.
131. **Nvidia.** *Whitepaper NVIDIA GF100*, Nvidia, 2009.
132. **Wittenbrink C., Kilgariff E. i Prabhu A.** *Fermi GF100 GPU Architecture*, IEEE Micro, vol. 31, no. 2, pp. 50-59, 2011.
133. **Sharma G. i Martin J.** *MATLAB: A language for Parallel Computing*, International Journal of Parallel Programming, vol. 37, no. 1, pp. 3-36, 2009.
134. **MathWorks.** *Parallel Computing Toolbox User's Guide*, 2013.

135. —. *Distributed Computing Toolbox 3 User's Guide*, 2007.
136. **Demmel J.** *Applied Numerical Linear Algebra*, Philadelphia, SIAM, 1997.
137. **Intel.** *Intel Math Kernel Library Reference Manual*, 2010.
138. **Advanced Micro Devices.** *AMD Core Math Library (ACML)*, 2008.
139. **NVIDIA.** *CUBLAS Library*, 2010.
140. **Anderson E. i inni.** *LAPACK Users' Guide. Third Edition*, Philadelphia, SIAM, 1999.
141. **EM Photonics Inc.** *CULA programmer's guide*, 2011.
142. —. *CULA reference manual*, 2011.
143. **Nath R., Tomov S. i Dongerra, J.** *An Improved MAGMA GEMM for Fermi GPUs*, International Journal of High Performance Computing, vol. 24, no. 4, pp. 511-515, 2010.
144. **Karlsson B.** *Beyond the C++ Standard Library. An Introduction to Boost*, Boston, Addison-Wesley, 2006.
145. **Åström K. i Bell R.** *Drum-boiler dynamics*, Automatica, 36, pp. 363-378, 2000.
146. **Flynn M. i O'Malley M.** *A drum boiler model for long term power system dynamic simulation*, IEEE Transactions on Power Systems, 14(1), pp. 209-217, 1999.
147. **Chaibakhsh A., Ghaffari A. i Moosavian S.** *A simulated model for a once-through boiler by parameter adjustment based on genetic algorithms*, Simulation Modelling Practice and Theory, 15(9), pp. 1029-1051, 2007.
148. **Latawiec K. i inni.** *Identification of a Boiler proper Using a Complex-Structure Model and Multivariable Orthonormal Basis Functions*, Proc. 12th IEEE International Conference on Methods and Models in Automation and Robotics, MMAR'2006, Międzyzdroje 28-31.08.2006; pp. 935-938, 2006.
149. **Stanisławski W. i Rydel M.** *Hierarchical mathematical models of complex plants on the basis of power boiler example*, Archives of Control Sciences, Vol. 20(LVI), No. 4, pp. 381-416, 2010.
150. **Stanisławski W. i Minkina W.** *Weryfikacja modelu matematycznego parownika kotła BP-1150 do celów sterowania*, Pomiary Automatyka Kontrola nr 3, pp. 7-10, 1999.
151. **Chen C.-T.** *Linear System Theory and Design. Third edition*, Oxford, Oxford University Press, 1999.
152. **Skogestad S. i Postletwaite I.** *Multivariable Feedback Control Analysis and design*, London, John Willey & Sons, 2005.
153. **Zieliński T.** *Cyfrowe przetwarzanie sygnałów. Od teorii do zastosowań*, Warszawa, WKŁ, 2009.
154. **Lieslehto J.** *PID controller tuning using evolutionary programming*, Proc of American Control Conference, vol. 4, pp. 2828-2833, 2001.
155. **Vandendorpe A. i Van Dooren P.** *Rozdział: Model reduction of interconnected systems, w: Schilders, W., Van der Vorst, H., Rommes, J.: Model order reduction: theory, research aspects and applications*, Berlin, Springer series Mathematics in Industry, vol. 13, pp. 305-321, 2008.
156. **Reis T. i Stykel T.** *Rozdział: A survey on model reduction of coupled systems, w: Schilders, W., Van der Vorst, H., Rommes, J.: Model order reduction: theory, research aspects and applications*, Berlin, Springer series Mathematics in Industry, vol. 13, pp. 133-155, 2008.

Spis ilustracji

Rysunek 1. Proces redukcji wymiaru modeli obiektów fizycznych	4
Rysunek 2. schemat blokowy realizacji procesu redukcji modeli złożonych z wykorzystaniem opracowanej metodologii	5
Rysunek 3. Operacja redukcji Rzędu modeli w przestrzeni stanów	6
Rysunek 4. Grafowa Reprezentacja systemów powiązanych	7
Rysunek 5. Struktura macierzy stanu A systemu, w którym wydzielono cztery podsystemy ...	7
Rysunek 6. wykorzystywane dekompozycje modeli LTI	7
Rysunek 7. schemat blokowy redukcji połączonej z dekompozycją	8
Rysunek 8. Graf dekompozycji wielopoziomowej	9
Rysunek 9. Struktura zastosowanego w pracy klastra obliczeniowego	10
Rysunek 10. Porównanie architektury CPU i GPU	10
Rysunek 11. Schemat blokowy algorytmu doprowadzającego gramian do postaci macierzy dodatnio określonej	16
Rysunek 12. Przykład dekompozycji fizycznej	21
Rysunek 13. sposób łączenia modeli zredukowanych dla dekompozycji strukturalnej uproszczonej	22
Rysunek 14. Modyfikacja macierzy B_1 i c_2	23
Rysunek 15. Modyfikacja macierzy B_2 i c_1	23
Rysunek 16. Łączenie modeli zredukowanych z uwzględnieniem zmodyfikowanych macierzy B i C	24
Rysunek 17. Struktura macierzy a po wykonaniu jednopoziomowej oraz dwupoziomowej dekompozycji epsilon	25
Rysunek 18. Macierz połączeń oraz wynikający z niej graf	26
Rysunek 19. Wartości własne modelu strefy dogrzewu dla 20 elementów skończonych oraz rzędu elementu skończonego 15	28
Rysunek 20. Binarna macierz połączeń dla modelu strefy dogrzewu, uzyskana z zastosowaniem klasycznego algorytmu dekompozycji Epsilon	28
Rysunek 21. Binarna macierz połączeń dla modelu strefy dogrzewu parownika, uzyskana na podstawie zmodyfikowanego algorytmu dekompozycji Epsilon	28
Rysunek 22. Graf powiązań zmiennych stanu dla zmodyfikowanej dekompozycji Epsilon. (kolorem czerwonym zaznaczono podsystem z dużymi wartościami bezwzględnymi wartości własnych (szybkie procesy dynamiczne), natomiast kolorem zielonym – podsystem wolnych procesów dynamicznych)	29
Rysunek 23. Wyznaczanie binarnej macierzy połączeń dla zmodyfikowanego algorytmu Epsilon	30
Rysunek 24. Zaproponowane dekompozycje slow-fast	30
Rysunek 25. Schemat łączenia modeli zredukowanych w wyniku dekompozycji mls	32
Rysunek 26. Schemat łączenia modeli zredukowanych w wyniku dekompozycji Schur complement	32
Rysunek 27. Schemat łączenia modeli zredukowanych z zastosowaniem zmodyfikowanej metody Schur complement (SchCompZm)	33
Rysunek 28. Schemat łączenia modeli zredukowanych w wyniku dekompozycji schur'a	34
Rysunek 29. Schemat łączenia modeli zredukowanych w wyniku dekompozycji schur-sylvester	35
Rysunek 30. rozważane kombinacje dla wielopoziomowych metod dekompozycyjnych	38
Rysunek 31. Dekompozycja dwupoziomowa	39
Rysunek 32. Przyjęta notacja oznaczania interfejsów	39
Rysunek 33. graf dekompozycji dwupoziomowej	40
Rysunek 34. modyfikacja interfejsów drugiego poziomu	40

Rysunek 35. Modyfikacja interfejsu pierwszego poziomu.....	40
Rysunek 36. konkatencja macierzy transformacji	40
Rysunek 37. Graf dekompozycji slow-fast na pierwszym poziomie dekompozycji	41
Rysunek 38. Graf dekompozycji Schur-Sylvester na pierwszym poziomie dekompozycji.....	42
Rysunek 39. Graf dekompozycji Schur-Sylvester na obydwu poziomach dekompozycji	42
Rysunek 40. Graf dekompozycji Schur-Sylvester na drugim poziomie dekompozycji (przekształcenie interfejsu pierwszego poziomu macierzami przekształceń dekompozycji) ...	42
Rysunek 41. Graf dekompozycji Schur-Sylvester na drugim poziomie dekompozycji (przekształcenie interfejsu pierwszego poziomu macierzami transformacji redukcji).....	42
Rysunek 42. Graf dekompozycji Schur'a na pierwszym poziomie dekompozycji (przekształcanie interfejsów macierzami transformacji redukcji)	43
Rysunek 43. Graf dekompozycji Schur'a na drugim poziomie dekompozycji (przekształcenie interfejsu pierwszego poziomu macierzami przekształceń dekompozycji).....	43
Rysunek 44. Graf dekompozycji Schur'a na drugim poziomie dekompozycji (przekształcanie interfejsów drugiego poziomu macierzami transformacji redukcji).....	43
Rysunek 45. Graf dekompozycji Schur'a na drugim poziomie dekompozycji (przekształcenie interfejsu pierwszego poziomu macierzami transformacji redukcji).....	44
Rysunek 46. Graf dekompozycji Schur complement na pierwszym poziomie dekompozycji	44
Rysunek 47. Graf dekompozycji Schur Complement na pierwszym poziomie dekompozycji (modyfikacja interfejsu poziomu pierwszego macierzami transformacji redukcji)	44
Rysunek 48. Graf dekompozycji Schur Complement na drugim poziomie dekompozycji.....	45
Rysunek 49. wydzielenie bloków reprezentujących cztery podsystemy w równaniach stanu modelu	45
Rysunek 50. Łączenie zredukowanych macierzy z wykorzystaniem dwupoziomowej dekompozycji opartej na dopełnieniu Schur'a.....	47
Rysunek 51. Graf dekompozycji strukturalnej na pierwszym poziomie dekompozycji	48
Rysunek 52. Graf dekompozycji Strukturalnej na drugim poziomie dekompozycji.....	48
Rysunek 53. graf zmodyfikowanej dekompozycji Schur Complement na pierwszym poziomie dekompozycji.....	49
Rysunek 54. Graf zmodyfikowanej dekompozycji Schur Complement na drugim poziomie dekompozycji.....	49
Rysunek 55. Architektury SM-MIMD (systemy z pamięcią współdzieloną)	51
Rysunek 56. architektura DM-MIMD (systemy z pamięcią rozproszoną).....	52
Rysunek 57. Typowa architektura klastrowa.....	52
Rysunek 58. Klastrowy komputerowy z procesorami graficznymi.....	52
Rysunek 59. Schemat realizacji zadania w klastrze matlab.....	54
Rysunek 60. Składnia polecenia <code>createTask</code>	55
Rysunek 61. Cykl życia zadań dla klastrowego środowiska Matlab	55
Rysunek 62. Zbiór modułów zaimplementowanej aplikacji (kolor szary – moduły zaimplementowane dla środowiska MATLAB, kolor zielony – moduły zaimplmentowane dla środowiska procesora graficznego, kolor pomarańczowy – wykorzystywane zewnętrzne programy).....	58
Rysunek 63. Schemat działania modułu zarządzania zadaniami klastra	59
Rysunek 64. Uproszczony schemat blokowy procesu redukcji modeli w środowisku klastrowym.....	61
Rysunek 65. Uproszczony schemat blokowy procesu dekompozycji	62
Rysunek 66. Uproszczony schemat blokowy scalania zredukowanych modeli	63
Rysunek 67. Struktura klastra obliczeniowego.....	64
Rysunek 68. Czas realizacji dziesięciu zadań redukcji dla poszczególnych komputerów wykorzystywanych w klastrze	64

Rysunek 69. Czas potrzebny na realizację dziesięciu zadań redukcji w zależności od konfiguracji klastra	65
Rysunek 70. Porównanie czasu potrzebnego na wykonanie zadania redukcji dla GPU i CPU	65
Rysunek 71. Uproszczony schemat technologiczny kotła BP-1150 w przekroju	69
Rysunek 72. Uproszczony schemat parownika kotła BP-1150	70
Rysunek 73. Uproszczony schemat przepływu wody i pary w parowniku kotła BP-1150	71
Rysunek 74. Model parownika	71
Rysunek 75. Struktura modelu parownika kotła BP-1150 (R1, R2 -rurociągi łączące, F - filtr)	72
Rysunek 76. Struktura modelu rur ekranowych komory paleniskowej parownika kotła BP-1150	73
Rysunek 77. Model strefowy parownika przepływowego przy ustalonym końcu strefy odparowania oraz podstawowe parametry czynnika roboczego.....	74
Rysunek 78. Element skończony modelu o parametrach skupionych.....	76
Rysunek 79. Dyskretyzacja ścianki rury oraz płetwy łączącej.....	77
Rysunek 80. Model elementu skończonego strefy dogrzewu w MATLAB/Simulink	81
Rysunek 81. Porównanie błędów modeli zredukowanych z wykorzystaniem różnych typów dekompozycji.....	85
Rysunek 82. Mapy stabilności modeli zredukowanych (kolor biały – model stabilny, kolor czarny – model niestabilny).....	87
Rysunek 83. Metody dekompozycji dwupoziomowych.....	88
Rysunek 84. przykładowe błędy aproksymacji modeli zredukowanych dla różnych metod dekompozycji dwupoziomowej	88
Rysunek 85. mapy stabilności modeli zredukowanych dla wybranych dekompozycji dwupoziomowych (kolor biały – model stabilny, kolor czarny – model niestabilny, kolor czerwony – model niezrealizowany z powodu błędów numerycznych)	90
Rysunek 86. Schemat łączenia modeli podsystemów rur ekranowych	91
Rysunek 87. Przykładowe wykresy charakterystyk częstotliwościowych dla modelu oryginalnego i modeli zredukowanych strefy dogrzewu (tor: $\Delta q^{\sim} \rightarrow \Delta h_{N1}$).....	93
Rysunek 88. Przykładowe wykresy charakterystyk częstotliwościowych dla modelu oryginalnego i modeli zredukowanych strefy dogrzewu (tor: $\Delta h_w \rightarrow \Delta h_{N1}$).....	94
Rysunek 89. Przykładowe wykresy bezwzględnego i względnego błędów aproksymacji dla modeli strefy dogrzewu (tor: $\Delta q^{\sim} \rightarrow \Delta h_{N1}$).....	94
Rysunek 90. Przykładowe wykresy bezwzględnego i względnego błędów aproksymacji dla modeli strefy dogrzewu (tor: $\Delta h_w \rightarrow \Delta h_{N1}$).....	94
Rysunek 91. Struktura macierzy A Po redukcji z zastosowaniem dekompozycji $Sch/SchSylv$	96
Rysunek 92. Struktura macierzy A Po redukcji z zastosowaniem dekompozycji Str/Str	96
Rysunek 93. Przykładowe wykresy charakterystyk częstotliwościowych dla modelu oryginalnego i modeli zredukowanych strefy odparowania I (tor: $\Delta q^{\sim} \rightarrow \Delta M_{II}$).....	96
Rysunek 94. Przykładowe wykresy charakterystyk częstotliwościowych dla modelu oryginalnego i modeli zredukowanych strefy odparowania I (tor: $\Delta h_{N1} \rightarrow \Delta h_{II}$).....	97
Rysunek 95. Przykładowe wykresy bezwzględnego i względnego błędów aproksymacji dla modeli strefy odparowania I (tor: $\Delta q^{\sim} \rightarrow \Delta M_{II}$).....	97
Rysunek 96. Przykładowe wykresy bezwzględnego i względnego błędów aproksymacji dla modeli strefy odparowania I (tor: $\Delta h_{N1} \rightarrow \Delta h_{II}$)	98

Rysunek 97. Przykładowe wykresy charakterystyk częstotliwościowych dla modelu oryginalnego i modeli zredukowanych strefy odparowania II (tor: $\Delta h_{II} \rightarrow \Delta h_{III}$).....	99
Rysunek 98. Przykładowe wykresy charakterystyk częstotliwościowych dla modelu oryginalnego i modeli zredukowanych strefy odparowania II (tor: $\Delta M_{we} \rightarrow \Delta M_{wy}$).....	99
Rysunek 99. Przykładowe wykresy bezwzględnego i względnego błędu aproksymacji dla modeli strefy odparowania II (tor: $\Delta h_{II} \rightarrow \Delta h_{III}$).....	100
Rysunek 100. Przykładowe wykresy bezwzględnego i względnego błędu aproksymacji dla modeli strefy odparowania II (tor: $\Delta M_{II} \rightarrow \Delta M_{III}$).....	100
Rysunek 101. Przykładowe wykresy charakterystyk częstotliwościowych dla modelu oryginalnego i modeli zredukowanych strefy odparowania III (tor: $\Delta h_{III} \rightarrow \Delta h_{SEP}$).....	102
Rysunek 102. Przykładowe wykresy charakterystyk częstotliwościowych dla modelu oryginalnego i modeli zredukowanych strefy odparowania III (tor: $\Delta P_{SEP} \rightarrow \Delta M_{SEP}$).....	102
Rysunek 103. Przykładowe wykresy bezwzględnego i względnego błędu aproksymacji dla modeli strefy odparowania III (tor: $\Delta h_{III} \rightarrow \Delta h_{SEP}$).....	102
Rysunek 104. Przykładowe wykresy bezwzględnego i względnego błędu aproksymacji dla modeli strefy odparowania III (tor: $\Delta P_{SEP} \rightarrow \Delta M_{SEP}$).....	103
Rysunek 105. Wykres wartości własnych macierzy stanu A modelu separatora.....	103
Rysunek 106. Wykres błędu redukcji w zależności od rzędu modelu zredukowanego dla modelu separatora.....	104
Rysunek 107. Przykładowe wykresy charakterystyk częstotliwościowych dla modelu oryginalnego i modeli zredukowanych separatora dla wielkości wejściowej ΔM_p (tor: $\Delta M_p \rightarrow \Delta P_{sep}$).....	104
Rysunek 108. Przykładowe wykresy charakterystyk częstotliwościowych dla modelu oryginalnego i modeli zredukowanych separatora dla wielkości wejściowej ΔM_p (tor: $\Delta M_p \rightarrow \Delta H_{sep}$).....	105
Rysunek 109. Wykres błędu redukcji w zależności od rzędu modelu zredukowanego dla modelu układu cyrkulacji.....	105
Rysunek 110. Przykładowe wykresy charakterystyk częstotliwościowych dla modelu oryginalnego i modeli zredukowanych układu cyrkulacji (tor: $\Delta P_{sep} \rightarrow \Delta h_w$).....	106
Rysunek 111. Przykładowe wykresy charakterystyk częstotliwościowych dla modelu oryginalnego i modeli zredukowanych układu cyrkulacji (tor: $\Delta h_{w_eko} \rightarrow \Delta h_w$).....	106
Rysunek 112. Wykresy błędu redukcji powtórnie zredukowanego modelu rur ekranowych, w stosunku do modelu pierwotnie zredukowanego w zależności od rzędu modelu powtórnie zredukowanego.....	107
Rysunek 113. Przykładowe wykresy charakterystyk częstotliwościowych dla modelu wstępnie zredukowanego i modeli powtórnie zredukowanych rur ekranowych (tor: $\Delta q \rightarrow \Delta M_{SEP}$).....	108
Rysunek 114. Przykładowe wykresy charakterystyk częstotliwościowych dla modelu wstępnie zredukowanego i modeli powtórnie zredukowanych rur ekranowych (tor: $\Delta h_w \rightarrow \Delta h_{SEP}$).....	108
Rysunek 115. Wykres wartości własnych macierzy stanu A wstępnie zredukowanego modelu parownika.....	109
Rysunek 116. Sprężenia zwrotne funkcjonujące w parowniku.....	109

Rysunek 117. Błąd modeli zredukowanych parownika w stosunku do modelu 183 rzędu (modelu odniesienia redukcji)	110
Rysunek 118. Błąd modeli zredukowanych parownika w stosunku do modelu wstępnie zredukowanego rzędu 2132 zmiennych stanu	110
Rysunek 119. Przykładowe wykresy charakterystyk częstotliwościowych dla modeli parownika (tor: $\Delta M_p \rightarrow \Delta H_{sep}$)	110
Rysunek 120 Przykładowe wykresy charakterystyk częstotliwościowych dla modeli parownika (tor: $\Delta q \rightarrow \Delta P_{sep}$)	111
Rysunek 121. Uproszczony schemat technologiczny układu przygotowania paliwa	114
Rysunek 122. Schemat układu regulacji parownika kotła BP-1150.....	115
Rysunek 123. schemat obiektu z obserwatorem stanu i sprzężeniem zwrotnym	116
Rysunek 124. Porównanie modułu transmitancji układu zamkniętego (stopień tłumienia zakłócenia ΔM_p) dla wybranych regulatorów wyznaczonych metodami: LQG, PLACE-LQRY, KALMAN-PLACE, PLACE-PLACE oraz PD wyznaczonym dla modelu rzędu 10120	
Rysunek 125. Porównanie modułu transmitancji układu zamkniętego (stopień tłumienia zakłócenia ΔM_p) dla wybranych regulatorów wyznaczonych metodami: LQG, PLACE-LQRY, KALMAN-PLACE, PLACE-PLACE oraz PD wyznaczonym dla modelu rzędu 15121	
Rysunek 126. Porównanie modułu transmitancji układu zamkniętego (stopień tłumienia zakłócenia ΔM_p) dla wybranych regulatorów wyznaczonych metodami: LQG, PLACE-LQRY, KALMAN-PLACE, PLACE-PLACE oraz PD wyznaczonym dla modelu rzędu 21121	
Rysunek 127. Charakterystyki skokowe układu zamkniętego dla zakłócenia ΔM_p	122
Rysunek 128. Schemat blokowy realizowanych działań, prowadzących do otrzymania zadawalającego jakościowo układu sterowania.....	128

Spis tabel

Tabela 1. Konfiguracja klastra.....	65
Tabela 2. summaryczne wyniki dla przebadanych metod dekompozycji jednopoziomowej w połączeniu z redukcją.....	86
Tabela 3. Summaryczne wyniki określające minimalny błąd modelu zredukowanego oraz procent modeli stabilnych dla rozważanych kombinacji dekompozycji	89
Tabela 4. Tabela wyników dla redukcji modelu strefy dogrzewu	93
Tabela 5. Tabela wyników dla redukcji modelu strefy odparowania I.....	95
Tabela 6. Tabela wyników dla redukcji strefy odparowania II	98
Tabela 7. Tabela wyników dla redukcji strefy odparowania III	101
Tabela 8. Podstawowe informacje na temat wybranych modeli wykorzystanych w celu uzyskania wstępnie zredukowanego modelu ruru ekranowych.....	107
Tabela 9. Podstawowe informacje na temat wybranych modeli wykorzystanych w celu uzyskania wstępnie zredukowanego modelu parownika	108
Tabela 10. Podstawowe parametry wyznaczonych układów sterowania	119
Tabela 11. Dobór nastaw regulatorów PD dla modeli rzędu 10, 15, 21	120

DODATEK A. ZBIÓR WYBRANYCH PROCEDUR LAPACK I CULA

Przedrostek x przed nazwą każdej procedury LAPACK oraz wewnątrz nazwy procedury CULA określa typ danych, na którym będą wykonywane operacje. Procedury biblioteki LAPACK/CULA zostały przygotowane w czterech wariantach, do wykonywania operacji na danych:

- rzeczywistych typu SINGLE (przedrostek S),
- zespolonych typu SINGLE (przedrostek C),
- rzeczywistych typu DOUBLE (przedrostek D),
- zespolonych typu DOUBLE (przedrostek Z).

Dodatkowo procedury CULA zostały przygotowane w dwóch wariantach. Procedury poprzedzone przedrostkiem cula wykonują operacje na danych znajdujących się w pamięci operacyjnej komputera (dla obliczeń na karcie graficznej w pierwszej kolejności kopiowane są do pamięci karty graficznej). Procedury poprzedzone przedrostkiem culaDevice operują na danych umieszczonych w pamięci karty graficznej.

LAPACK	CULA	FUNKCJONALNOŚĆ
xGESV	culaxGESV, culaDevicexGESV	Rozwiązanie układu równań $AX=B$
xGETRF	culaxGETRF, culaDevicexGETRF	Faktoryzacja LU macierzy
xGETRI	culaxGETRI, culaDevicexGETRI	Odwrotność macierzy
xGETRS	culaxGETRS, culaDevicexGETRS	Rozwiązanie układu równań liniowych $AX=B$, $A^T X=B$ lub $A^* X=B$ z wykorzystaniem faktoryzacji LU
xPOSV	culaxPOSV, culaDevicexPOSV	Rozwiązanie układów równań liniowych $AX=B$ dla symetrycznej, dodatnio określonej macierzy A
xPOTRF	culaxPOTRF, culaDevicexPOTRF	Dekompozycja Cholesky'ego symetrycznej, dodatnio określonej macierzy
xPOTRS	culaxPOTRS, culaDevicexPOTRS	Rozwiązanie układu równań liniowych $AX=B$ z wykorzystaniem dekompozycji Cholesky'ego
xTRTRS	culaxTRTRS, culaDevicexTRTRS	Rozwiązanie układu równań liniowych $AX=B$, gdzie A jest macierzą trójkątną
xTRTRI	culaxTRTRI, culaDevicexTRTRI	Odwrotność macierzy trójkątnej
xGBTRF	culaxGBTRF, culaDevicexGBTRF	Faktoryzacja LU macierzy
xGEQRF	culaxGEQRF, culaDevicexGEQRF	Dekompozycja QR macierzy prostokątnej
xORGQR xUNGQR	culaxORGQR, ulaDevicexORGQR/ culaxUNGQR, culaDevicexUNGQR	Wyznaczenie ortogonalnej/unitarnej macierz Q z dekompozycji QR przeprowadzonej z wykorzystaniem procedury xGEQRF
xORMQR xUNMQR	culaxORMQR, culaDevicexORMQR/ culaxUNMQR, culaDevicexUNMQR	Iloczyn macierzy z ortogonalną/unitarną macierzą Q (z dekompozycji QR) uzyskaną w wyniku wykonania procedury xGEQRF
xGELQF	culaxGELQF, culaDevicexGELQF	Dekompozycja LQ macierzy
xORGLQ xUNGLQ	culaxORGLQ, culaDevicexORGLQ/ culaxUNGLQ, culaDevicexUNGLQ	Wyznaczenie ortogonalnej/unitarnej macierz Q z dekompozycji LQ wykonanej za pomocą xGELQF
xORMLQ xUNMLQ	culaxORMLQ, culaDevicexORMLQ/ culaxUNMLQ, culaDevicexUNMLQ	Iloczyn macierzy z ortogonalną/unitarną macierzą Q (z dekompozycji LQ) wyznaczoną z procedury xGELQF
xGERQF	culaxGERQF, culaDevicexGERQF	Dekompozycja RQ macierzy prostokątnej
xGGRQF	culaxGGRQF, culaDevicexGGRQF	Uogólniona dekompozycja RQ pary macierzy A i B
xORGRQ xUNGRQ	culaxORGRQ, culaDevicexORGRQ/ culaxORGRQ, culaDevicexORGRQ	Wyznaczenie ortogonalnej/unitarnej macierzy Q z dekompozycji RQ wykonanej za pomocą xGERQF
xORMRQ xUNMRQ	culaxORMRQ, culaDevicexORMRQ/ culaxUNMRQ, culaDevicexUNMRQ	Iloczyn macierzy z ortogonalną/unitarną macierzą Q (z dekompozycji RQ) wyznaczoną z procedury xGERQF
xGEQLF	culaxGEQLF, culaDevicexGEQLF	Dekompozycja QL macierzy prostokątnej
xORGQL xUNGQL	culaxORGQL, culaDevicexORGQL/ culaxUNGQL, culaDevicexUNGQL	Wyznaczenie ortogonalnej/unitarnej macierzy Q z dekompozycji QL wykonanej z wykorzystaniem procedury xGEQLF
xORMQL xUNMQL	culaxORMQL, culaDevicexORMQL/ culaxUNMQL, culaDevicexUNMQL	Iloczyn macierzy z ortogonalną/unitarną macierzą Q (z dekompozycji QL) wyznaczoną z procedury xGELQF
xGELS	culaxGELS, culaDevicexGELS	Rozwiązanie układu równań $AX=B$, $A^T X=B$ lub $A^* X=B$ wykorzystując metodę najmniejszych kwadratów

LAPACK	CULA	FUNKCJONALNOŚĆ
xORGHR xUNGHR	culaxORGHR, culaDevicexORGHR/ culaxUNGHR, culaDevicexUNGHR	Wyznaczenie ortogonalnej/unitarnej macierzy transformacji doprowadzającej macierz do postaci Hessenberg'a z wykorzystaniem procedury xGEHRD
xGESVD	culaxGESVD, culaDevicexGESVD	Dekompozycja SVD macierzy prostokątnej
xGEBRD	culaxGEBRD, culaDevicexGEBRD	Redukcja macierzy do postaci dwudiagonalnej z wykorzystaniem transformacji $Q^TAP=B$
xORGBR xUNGBR	culaxORGBR, culaDevicexORGBR/ culaxUNGBR, culaDevicexUNGBR	Wyznaczenie ortogonalnych/unitarnych macierzy transformacji Q i P przekształcających macierz do postaci dwudiagonalnej z wykorzystaniem procedury xGEBRD
xBDSQR	culaxBDSQR, culaDevicexBDSQR	Dekompozycja SVD macierzy dwudiagonalnej
xGGLSE	culaxGGLSE, culaDevicexGGLSE	Rozwiązanie układu równań z wykorzystaniem metody najmniejszych kwadratów i dekompozycji GRQ
xSYEV xHEEV	culaxSYEV, culaDevicexSYEV/ culaxHEEV, culaDevicexHEEV	Wyznaczenie wektorów i wartości własnych macierzy symetrycznej/Hermitowskiej
xSYEVX xHEEVX	culaxSYEVX, culaDevicexSYEVX/ culaxHEEVX, culaDevicexHEEVX	Wyznaczenie wybranych wektorów i wartości własnych macierzy symetrycznej/Hermitowskiej
xSTEBZ	culaxSTEBZ, culaDevicexSTEBZ	Wyznaczenie wybranych wartości własnych symetrycznej macierzy rzeczywistej trójdzielnej
xSTEQR	culaxSTEQR, culaDevicexSTEQR	Wyznaczenie wektorów i wartości własnych symetrycznej macierzy rzeczywistej trójdzielnej
xGEEV	culaxGEEV, culaDevicexGEEV	Wyznaczenie wektorów i wartości własnych macierzy
xGEHRD	culaxGEHRD, culaDevicexGEHRD	Wyznaczenie postaci Hessenberg'a macierzy

DODATEK B. Wybrane funkcje i procedury MATLAB związane z dekompozycją i redukcją równoległą

Funkcje i procedury środowiska MATLAB związane z dekompozycją i redukcją równoległą pojedynczego modelu	
Nazwa	Opis
dist_multi_dec_fin	<p>Parametry:</p> <p>path – ścieżka do katalogu zawierającego pliki z macierzami A,B,C modelu, methodss – wektor zawierający typ dekompozycji dla poszczególnych poziomów, ns – wektor określający na ile podsystemów należy zdekomponować model na kolejnych poziomach, k – docelowy rozmiar modelu zredukowanego (w przypadku dekompozycji slow-fast rozmiar części wolnej modelu zredukowanego), om_min - dolny zakres aproksymacji charakterystyk częstotliwościowych modelu zredukowanego (w przypadku dekompozycji slow-fast parametr określa dolny zakres aproksymacji charakterystyk częstotliwościowych części wolnej modelu zredukowanego), om_max – górny zakres aproksymacji charakterystyk częstotliwościowych modelu zredukowanego (w przypadku dekompozycji slow-fast parametr określa górny zakres aproksymacji charakterystyk częstotliwościowych dla części wolnej modelu zredukowanego), om_min_fast – dolny zakres aproksymacji charakterystyk częstotliwościowych części szybkiej modelu zredukowanego, om_max_fast – górny zakres aproksymacji charakterystyk częstotliwościowych części szybkiej modelu zredukowanego, k_fast – docelowy rozmiar części szybkiej modelu zredukowanego.</p> <p>Wyniki:</p> <p>kod_bledu – wynik określa czy funkcja została wykonana poprawnie.</p> <p>Opis:</p> <p>Funkcja wykonuje wielopoziomową dekompozycję oraz redukcję modelu oryginalnego z wykorzystaniem obliczeń równoległych. Poszczególne kroki wykonywane w funkcji mogą zostać opisane następująco:</p> <ol style="list-style-type: none"> dla przypadku dekompozycji wielopoziomowej nie związanej z dekompozycją

	<p>slow-fast, lub w przypadku dekompozycji jednopoziomowej wywoływana jest procedura <code>multi_dec</code> w celu dekompozycji modelu oryginalnego.</p> <p>1.1. W przypadku dekompozycji wielopoziomowej wyznaczane są ścieżki dla modeli najniższego poziomu (<code>generate_all_reduct_paths</code>), po czym wykonywana jest ich równoległa redukcja (<code>dist_reduct_f_all</code>). Wyznaczone macierze transformacji wykorzystywane są do przekształcenia interfejsów modeli wyższego poziomu dekompozycji. Przywracane są również zerowe kolumny macierzy B i zerowe wiersze macierzy C usunięte dla celów redukcji. Następuje proces scalania modeli (<code>propagate_down</code>).</p> <p>1.2. W przypadku dekompozycji jednopoziomowej redukowane są modele zdekomponowane (<code>dist_reduct_f</code>). W kolejnym kroku następuje proces scalania (<code>transform_old</code>).</p> <p>2. dla przypadku dekompozycji wielopoziomowej, gdzie na pierwszym poziomie realizowana jest dekompozycja slow-fast, w pierwszej kolejności model zdekomponowany jest na część szybką i część wolną (<code>multi_dec(path, methodss(1), ns(1))</code>). Następnie redukowana jest część szybka (<code>dist_reduct_f_fast</code>). W kolejnym kroku wykonywana jest dalsza dekompozycja oraz redukcja części wolnej modelu, poprzez rekurencyjne wywołanie funkcji <code>dist_multi_dec_fin</code>. Macierze stanu dla części wolnej modelu, oraz wynikowe macierze transformacji wyznaczane są za pomocą funkcji <code>transform_zm_slow</code>. W kolejnym kroku przywracane są zerowe wiersze macierzy C lub zerowe kolumny macierzy B usunięte dla potrzeb redukcji (<code>veryfy_matrix_2, veryfy_matrix_1</code>). Ostateczny proces scalania realizowany jest za pomocą funkcji <code>transform_sf</code>.</p>
<p><code>multi_dec</code></p>	<p>Parametry:</p> <p><code>path</code> – ścieżka do katalogu zawierającego pliki z macierzami A,B,C modelu, <code>methodss</code> - wektor zawierający typ dekompozycji dla poszczególnych poziomów, <code>ns</code> - wektor określający na ile podsystemów należy zdekomponować model na kolejnych poziomach,</p> <p>Opis:</p> <p>Procedura tworzy hierarchię plików i folderów realizując wielopoziomową dekompozycję modelu. Ścieżki dla poszczególnych poziomów dekompozycji modelu wyznaczane są poprzez wywołanie funkcji <code>generate_paths</code>. Dekompozycja modelu wykonywana jest poprzez wywołanie procedury <code>dec_f</code>.</p>
<p><code>generate_paths</code></p>	<p>Parametry:</p> <p><code>path</code> - ścieżka do katalogu zawierającego pliki z macierzami A,B,C modelu, <code>methods</code> - wektor zawierający typ dekompozycji dla poszczególnych poziomów, <code>ns</code> - wektor określający na ile podsystemów należy zdekomponować model na kolejnych poziomach, <code>level</code> – poziom, dla którego zostaną wygenerowane ścieżki dla modeli zdekomponowanych.</p> <p>Wyniki:</p> <p><code>opath</code> – tablica komórek (ang. cell array) przechowująca ścieżki do modeli zdekomponowanych dla żadanego poziomu dekompozycji.</p> <p>Opis:</p> <p>Funkcja zwraca ścieżki do folderów zawierających macierze stanu modeli dla żadanego poziomu dekompozycji. Nazwy folderów zawierających modele zdekomponowane dla poszczególnych poziomów dekompozycji uzyskiwane są poprzez wywołanie funkcji <code>name of dec</code>.</p>
<p><code>generate_all_reduct_paths</code></p>	<p>Parametry:</p> <p><code>path</code> - ścieżka do katalogu zawierającego pliki z macierzami A,B,C modelu, <code>methods</code> - wektor zawierający typ dekompozycji dla poszczególnych poziomów, <code>ns</code> - wektor określający na ile podsystemów należy zdekomponować model na kolejnych poziomach, <code>level</code> – najniższy poziom dekompozycji określony liczbowo.</p> <p>Wyniki:</p> <p><code>opath</code> – tablica komórek (ang. cell array) przechowująca ścieżki dla modeli zdekomponowanych na najniższym poziomie dekompozycji.</p>

	<p>Opis: Funkcja zwraca ścieżki dla folderów zawierających macierze stanu modeli na najniższym poziomie dekompozycji.</p>
cell_2_str	<p>Parametry: cell – tablica komórek, i – liczba kolumn tablicy cell .</p> <p>Wyniki: str – ciąg znaków.</p> <p>Opis: Funkcja konwertuje tablicę komórek (ang. cell array) na ciąg znaków typu char.</p>
compute_ranks	<p>Parametry: types - wektor zawierający typ dekompozycji dla poszczególnych poziomów, ns - wektor określający na ile podsystemów należy zdekomponować model na kolejnych poziomach, k – docelowy rozmiar całego modelu zredukowanego.</p> <p>Wyniki: rzad – docelowy rząd modelu zredukowanego dla najniższego poziomu dekompozycji.</p> <p>Opis: Funkcja wyznacza rząd modelu zredukowanego dla najniższego poziomu dekompozycji tak, aby po połączeniu wszystkich zredukowanych modeli uzyskać docelowy rząd całego modelu.</p>
dist_reduct_f_all	<p>Parametry: type – typ dekompozycji dla najniższego poziomu, paths – tablica zawierająca ścieżki do folderów zawierających macierze stanu modeli najniższego poziomu dekompozycji, n – liczba modeli wydzielonych dla najniższego poziomu dekompozycji, k – docelowy rozmiar modelu zredukowanego (w przypadku dekompozycji slow-fast rozmiar części wolnej modelu zredukowanego), om_min - dolny zakres aproksymacji charakterystyk częstotliwościowych modelu zredukowanego (w przypadku dekompozycji slow-fast parametr określa dolny zakres aproksymacji charakterystyk częstotliwościowych części wolnej modelu zredukowanego), om_max – górny zakres aproksymacji charakterystyk częstotliwościowych modelu zredukowanego (w przypadku dekompozycji slow-fast parametr określa górny zakres aproksymacji charakterystyk częstotliwościowych dla części wolnej modelu zredukowanego), om_min_fast – dolny zakres aproksymacji charakterystyk częstotliwościowych części szybkiej modelu zredukowanego, om_max_fast – górny zakres aproksymacji charakterystyk częstotliwościowych części szybkiej modelu zredukowanego, k_fast – docelowy rozmiar części szybkiej modelu zredukowanego.</p> <p>Opis: Procedura realizuje redukcję modeli najniższego poziomu dekompozycji dla przypadku dekompozycji wielopoziomowej nie związanej z dekompozycją slow-fast.</p>
propagate_down	<p>Parametry: path – ścieżka do katalogu zawierającego pliki z macierzami A,B,C modelu, methods – wektor zawierający typ dekompozycji dla poszczególnych poziomów, ns – wektor określający na ile podsystemów należy zdekomponować model na kolejnych poziomach, native – parametr określający czy bieżące wywołanie procedury jest rekursywne (0 – nie, 1 – tak).</p> <p>Opis: Procedura realizuje przekształcanie oraz łączenie modeli począwszy od najniższego poziomu dekompozycji do poziomu najwyższego (scalanie). Przed wywołaniem procedury należy zredukować modele znajdujące się na najniższym poziomie dekompozycji oraz wyznaczyć macierze transformacji.</p> <p>1. Dla dekompozycji wielopoziomowej pierwsze wywołanie procedury (native=0) związane jest z transformacją macierzy modeli i interfejsów znajdujących się na</p>

	<p>najniższym poziomie dekompozycji (n). Proces ten realizowany jest poprzez wywołanie funkcji <code>transform</code>. Przy rekursywnym wywołaniu procedury (<code>native=1</code>) za realizację procesu scalania odpowiada procedura <code>transform_zm</code>.</p> <p>W wyniku działania obydwu procedur uzyskane zostają macierze stanu modeli oraz macierze transformacji interfejsów dla poziomu dekompozycji n-1. Kopiowane są one do folderów związanych z poziomem dekompozycji n-1 przy wykorzystaniu procedury <code>copy_right</code>. W kolejnym kroku następuje przywrócenie zerowych wierszy lub zerowych kolumn w macierzach B i C, usuniętych dla potrzeb redukcji, przy wykorzystaniu procedury <code>verify_matrix</code>.</p> <p>2. Dla dekompozycji jednopoziomowej (lub podczas ostatniego rekurencyjnego wywołania procedury) macierze podsystemów scalane są z wykorzystaniem procedury <code>transform_zm2</code>.</p>
<code>get_sys_1</code>	<p>Parametry: <code>path</code> – ścieżka do katalogu zawierającego pliki z macierzami A,B,C modelu, <code>type</code> – typ dekompozycji wykonywany na najwyższym poziomie.</p> <p>Opis: Procedura tworzy strukturę ss zredukowanego modelu całości, zapisując ją do folderu określonego parametrem <code>path</code>.</p>
<code>dist_reduct_f</code>	<p>Parametry: <code>type</code> – typ dekompozycji, <code>pahts</code> – ścieżka do folderu zawierającego macierze stanu modeli zdekomponowanych, <code>n</code> – liczba modeli wydzielonych w procesie dekompozycji, <code>k</code> – docelowy rozmiar modelu zredukowanego, <code>om_min</code> – dolny zakres aproksymacji charakterystyk częstotliwościowych modelu zredukowanego, <code>om_max</code> – górny zakres aproksymacji charakterystyk częstotliwościowych modelu zredukowanego.</p> <p>Opis: Procedura realizuje redukcję modelu zdekomponowanego dla przypadku dekompozycji jednopoziomowej nie związanej z dekompozycją slow-fast.</p>
<code>transform_old</code>	<p>Parametry: <code>type</code> – typ dekompozycji, <code>pahts</code> – ścieżka do folderu zawierającego macierze stanu modeli zdekomponowanych, <code>n</code> – liczba modeli wydzielonych w procesie dekompozycji.</p> <p>Opis: Procedura scala zdekomponowany (w wyniku dekompozycji jednopoziomowej) i zredukowany model.</p>
<code>dist_reduct_f_fast</code>	<p>Parametry: <code>type</code> – typ dekompozycji slow-fast, <code>path</code> – ścieżka do katalogu zawierającego pliki z macierzami A,B,C modelu, <code>om_min_fast</code> – dolny zakres aproksymacji charakterystyk częstotliwościowych części szybkiej modelu zredukowanego, <code>om_max_fast</code> – górny zakres aproksymacji charakterystyk częstotliwościowych części szybkiej modelu zredukowanego, <code>k_fast</code> – docelowy rozmiar części szybkiej modelu zredukowanego.</p> <p>Opis: Procedura realizuje redukcję części szybkiej modelu zdekomponowanego.</p>
<code>transform_zm_slow</code>	<p>Parametry: <code>path</code> – ścieżka do folderu zawierającego hierarchię folderów związaną z dekompozycją części wolnej modelu, <code>type</code> – typ dekompozycji realizowany na pierwszym poziomie dla części wolnej modelu, <code>n</code> – liczba modeli, na które została podzielona część wolna modelu na pierwszym poziomie dekompozycji.</p> <p>Wyniki: <code>t, l</code> – macierze transformacji (wynikające z połączenia wszystkich macierzy</p>

	<p>transformacji uzyskanych w procesie redukcji modeli najniższego poziomu części wolnej), a, b, c – macierze stanu zredukowanej części wolnej modelu. Opis: Funkcja łączy zdekomponowane i zredukowane modele części wolnej. W przypadku, gdy część wolna została zdekomponowana na większą liczbę poziomów niż jeden, przed wywołaniem funkcji następuje rekurencyjne wywołanie funkcji <code>dist_multi_dec_fin</code>. Zwracane przez funkcję macierze transformacji zostaną wykorzystane do przekształcenia interfejsu łączącego część szybką i część wolną.</p>
<code>copy_slow</code>	<p>Parametry: <code>path</code> - ścieżka do katalogu związanego z realizacją dekompozycji pierwszego poziomu, <code>type_main</code> – typ dekompozycji realizowany na pierwszym poziomie, t, l, a, b, c – macierze transformacji oraz macierze stanu części wolnej modelu, uzyskane w wyniku wywołania funkcji <code>transform_zm_slow</code>. Opis: Procedura kopiuje macierze stanu części wolnej oraz macierze transformacji do katalogu danej metody dekompozycji <code>slow-fast</code>.</p>
<code>verify_matrix_2</code>	<p>Parametry: <code>path</code> - ścieżka do katalogu związanego z realizacją dekompozycji pierwszego poziomu, <code>type</code> – typ dekompozycji realizowany na pierwszym poziomie, Opis: Procedura przywraca w części wolnej modelu zerowe wiersze macierzy C lub zerowe kolumny macierzy B, które zostały usunięte dla celów redukcji. Procedura wykorzystuje w tym celu funkcje <code>restore_b</code> oraz <code>restore_c</code>.</p>
<code>transform_sf</code>	<p>Parametry: <code>path</code> – ścieżka do katalogu związanego z realizacją dekompozycji <code>slow-fast</code>, <code>type</code> – rodzaj dekompozycji. Opis: Procedura realizuje scalanie części wolnej modelu.</p>
<code>dec_f</code>	<p>Parametry: <code>path</code> – ścieżka do folderu zawierającego pliki z macierzami stanu modelu zdekomponowanego, <code>type</code> – rodzaj wykonywanej dekompozycji, n – liczba modeli wydzielonych w procesie dekompozycji. Opis: Procedura realizuje dekompozycję modelu.</p>
<code>dec_str_full</code> , <code>dec_my_dec</code> , <code>dec_schc</code> , <code>dec_str_upr</code> , <code>dec_schc_upr2</code> , <code>dec_schur_dec</code>	<p>Parametry: <code>path</code> – ścieżka do folderu zawierającego pliki z macierzami stanu modelu, n – wektor zawierający rozmiar wydzielanych modeli w procesie dekompozycji. Opis: Procedury realizujące dekompozycje: <i>Str</i>, <i>SchSylv</i>, <i>SchComp</i>, <i>StrUp</i>, <i>SchCompZm</i>, <i>Sch</i>. Model dzielony jest na części, których rozmiar określony jest elementami wektora n.</p>
<code>dec_slow_fast</code> , <code>dec_slow_fast2</code> , <code>dec_slow_fast3</code> , <code>dec_slow_fast4</code> , <code>dec_slow_fast6</code> , <code>dec_slow_fast12</code> , <code>dec_slow_fast13</code> , <code>dec_slow_fast14</code>	<p>Parametry: <code>path</code> – ścieżka do folderu zawierającego pliki z macierzami stanu modelu, Opis: Procedury realizują dekompozycję <code>slow-fast</code> za pomocą zmodyfikowanej dekompozycji Epsilon. Dobierany jest automatycznie rozmiar poszczególnych części w taki sposób, aby w jednym modelu zgrupowane zostały zmienne stanu związane z procesami szybkimi, w drugim natomiast z procesami wolnymi. Po zastosowaniu wyznaczonych wektorów permutacji oba modele rozdzielane są za pomocą dekompozycji (w zależności od procedury): <i>Str</i>, <i>StrUp</i>, <i>MLS</i>, <i>SchComp</i>, <i>SchSylv</i>, <i>Sch</i>, <i>SchCompZm</i>, MATLAB <code>slowfast</code>.</p>
<code>size_of_a</code>	<p>Opis: Funkcja zwraca rozmiar macierzy stanu A modelu, której plik jest zapisany w bieżącej lokalizacji.</p>

calc_orders	<p>Parametry: a – rząd macierzy stanu A, n – liczba modeli najniższego poziomu, wyznaczanych w procesie dekompozycji.</p> <p>Wyniki: orders – wektor zawierający rozmiary podsystemów.</p> <p>Opis: Funkcja na podstawie rzędu macierzy stanu A oraz liczby żądanych podsystemów najniższego poziomu tworzy wektor zawierający rozmiar podsystemów.</p>
matrix_of_possible	<p>Parametry: v – wektor określający liczbę wydzielonych podsystemów dla poszczególnych poziomów dekompozycji,</p> <p>Wyniki: all – ścieżki folderów dla plików podsystemów najniższego poziomu.</p> <p>Opis: Redukcja zdekomponowanego modelu wiąże się z redukcją modeli wydzielonych na najniższym poziomie dekompozycji. Funkcja określa ścieżki folderów dla wszystkich modeli wydzielonych na najniższym poziomie dekompozycji.</p>
pre_number, post_number	<p>Parametry: v - wektor określający liczbę wydzielonych podsystemów dla poszczególnych poziomów dekompozycji, i – poziom dekompozycji.</p> <p>Wyniki: n – macierz zawierająca ścieżki do folderów podsystemów wydzielonych na poziomie dekompozycji i-1 (pre_number) lub i+1 (post_number)</p> <p>Opis: Funkcja wyznacza macierz zawierającą ścieżki do folderów nadrzędnych (pre_number) lub podrzędnych (post_number) względem poziomu i.</p>
name_of_dec	<p>Parametry: dec – typ dekompozycji.</p> <p>Wyniki: name – nazwa dekompozycji (ciąg znaków typu char).</p> <p>Opis: Funkcja na podstawie przekazanej liczby zwraca ciąg znaków określających nazwę dekompozycji (element wykorzystywany przy operacjach na hierarchii plików).</p>
dist_str_full_reduct_all, dist_my_dec_reduct_all, dist_mls_reduct_all, dist_str_upr_reduct_all, dist_mls_reduct_all, dist_slow_fast_reduct_all, dist_slow_fast_reduct2, dist_slow_fast_reduct3, dist_schur_dec_full_reduct_all	<p>Parametry: paths – tablica komórek (ang. cell array) zawierająca ścieżki do podsystemów najniższego poziomu dekompozycji, n – liczba wydzielonych podsystemów najniższego poziomu, om_min - dolny zakres aproksymacji charakterystyk częstotliwościowych modelu zredukowanego, om_max – górny zakres aproksymacji charakterystyk częstotliwościowych modelu zredukowanego, k – docelowy rozmiar modelu zredukowanego.</p> <p>Opis: Zbiór procedur realizujących równoległą redukcję zdekomponowanego modelu dla przypadku dekompozycji wielopoziomowej nie związanej z dekompozycją slow-fast. Procedury realizują następujące operacje:</p> <ol style="list-style-type: none"> 1. Wyszukują menedżera zadań klastra MATLAB. 2. Tworzą nowe zadanie (ang. job). 3. Określają zależności folderów pomiędzy poszczególnymi węzłami klastra. 4. Sprawdzają poprawność zredukowanych modeli (verify_sys). 5. Tworzą wątki (ang. tasks) związane z redukcją poszczególnych modeli znajdujących się na najniższym poziomie dekompozycji (funkcja dist_reduct). Dla każdego wątku przekazywany jest model przeznaczony do redukcji oraz parametry redukcji. 6. Umieszczają zdefiniowane zadanie (z przydzielonymi wątkami) w kolejce zadań gotowych do realizacji. 7. Oczekują na zakończenie wątków. 8. Pobierają wyniki zakończonych wątków (macierze modelu zredukowanego oraz

	macierze transformacji). 9. Zachowują pobrane wyniki.
verify_sys	Parametry: sys – model w postaci obiektu ss. Wyniki: x – wartość określająca czy dany model jest poprawny (1 – model poprawny, 0 – model błędny). Opis: Funkcja weryfikuje przekazany model w celu określenia jego poprawności. Sprawdzeniu podlega między innymi rząd macierzy stanu B i C i stabilność modelu.
dist_reduct	Parametry: sys – model do redukcji w postaci obiektu ss, om_min - dolny zakres aproksymacji charakterystyk częstotliwościowych modelu zredukowanego, om_max – górny zakres aproksymacji charakterystyk częstotliwościowych modelu zredukowanego, om_lp – liczba punktów, w których zostanie wyznaczony błąd modelu zredukowanego, k_min – minimalny rozmiar modelu zredukowanego, k_max – maksymalny rozmiar modelu zredukowanego, iter – liczba iteracji dla każdego rzędu modelu zredukowanego. Wyniki: rsys – model zredukowany w postaci obiektu ss, macierz_transformacji_T, macierz_transformacji_IT – macierze transformacji T i L. Opis: Funkcja redukuje model sys z wykorzystaniem zadanych parametrów redukcji.
str_from_clock	Opis: Procedura generuje unikalną nazwę, (nazwa dla folderów tymczasowych).
dec_MP	Parametry: name – nazwa modelu, path – ścieżka do pliku modelu zachowanego w postaci obiektu ss, om_min - dolny zakres aproksymacji charakterystyk częstotliwościowych modelu zredukowanego, om_max – górny zakres aproksymacji charakterystyk częstotliwościowych modelu zredukowanego, om_lp – liczba punktów, dla których zostanie wyznaczony błąd modelu zredukowanego. Opis: Procedura tworzy strukturę MP zawierającą opis modelu oryginalnego i parametry dla wyznaczania błędów redukcji. Struktura jest niezbędna dla wywołania przybownika eAMOR
dec_AE	Parametry: source_path – ścieżka do pliku zawierającego szablon struktury AE, path – ścieżka określająca folder zapisu utworzonej struktury AE, k_min – minimalny rozmiar modelu zredukowanego, k_max – maksymalny rozmiar modelu zredukowanego, iter – liczba iteracji dla każdego rzędu modelu zredukowanego. Opis: Procedura tworzy strukturę AE zawierającą parametry dla ewolucyjnego algorytmu doboru parametrów filtrów dla metody redukcji FW.
dec_PIP	Parametry: path - ścieżka określająca folder zapisu struktury PIP, MP – struktura opisująca model oryginalny, AE – struktura zawierająca parametry dla przybownika eAMOR. Opis: Procedura tworzy strukturę PIP zawierającą parametry dla filtrów wykorzystywanych przy redukcji FW.

transform	<p>Parametry:</p> <p>path – ścieżka do folderu najniższego poziomu dekompozycji, type – typ dekompozycji realizowany na najniższym poziomie dekompozycji, n – liczba podsystemów, na które został podzielony model na najniższym poziomie dekompozycji.</p> <p>Wyniki:</p> <p>t, l – macierze transformacji dla poziomu wyższego, a, b, c – macierze stanu modeli podsystemów dla poziomu wyższego,</p> <p>Opis:</p> <p>Funkcja scala zdekomponowane i zredukowane modele znajdujące się na najniższym poziomie dekompozycji. Jej wyniki stanowią macierze stanu oraz macierze transformacji dla poziomu wyższego.</p>
transform_zm	<p>Parametry:</p> <p>path – ścieżka folderu dla i-tego poziomu dekompozycji, type – typ dekompozycji realizowany na i-tym poziomie dekompozycji, n – liczba podsystemów, na które został podzielony model na i-tym poziomie dekompozycji.</p> <p>Wyniki:</p> <p>t, l – macierze transformacji wyznaczone dla poziomu wyższego, a, b, c – macierze stanu modeli podsystemów dla poziomu wyższego</p> <p>Opis:</p> <p>Funkcja scala zdekomponowane i zredukowane modele znajdujące się na i-tym poziomie dekompozycji. Jej wyniki stanowią macierze stanu oraz macierze transformacji dla poziomu i-1.</p>
copy_right	<p>Parametry:</p> <p>type – typ dekompozycji na poziomie i-1, t, l, a, b, c – macierze stanu oraz macierze transformacji wyznaczone przez scalanie zdekomponowanych części na poziomie i-tym.</p> <p>Opis:</p> <p>Procedura kopiuje do folderów poziomu i-1 scalone macierze transformacji i macierze stanu modeli uzyskanych na poziomie i-tym</p>
verify_matrix	<p>Parametry:</p> <p>path – ścieżki modeli dla poziomu i-1, type – typ dekompozycji dla poziomu i-1, ns – liczba wydzielonych podsystemów dla poziomu i-1.</p> <p>Opis:</p> <p>Procedura przywraca zerowe kolumny macierzy B lub zerowe wiersze macierzy C dla wszystkich modeli znajdujących się na poziomie dekompozycji i-1 (elementy usunięte dla potrzeb redukcji).</p>
restore_b	<p>Parametry:</p> <p>b – macierz wejść B, vec – wektor zawierający informację o usuniętych kolumnach macierzy B.</p> <p>Wyniki:</p> <p>x – macierz wejść B z przywróconymi zerowymi kolumnami.</p> <p>Opis:</p> <p>Funkcja przywraca usunięte zerowe kolumny macierzy wejść B, które zostały usunięte dla celów redukcji.</p>
restore_c	<p>Parametry:</p> <p>c – macierz wyjść C, vec – wektor zawierający informację o usuniętych wierszach macierzy C.</p> <p>Wyniki:</p> <p>x – macierzy wyjść C z przywróconymi zerowymi wierszami.</p> <p>Opis:</p> <p>Funkcja przywraca usunięte zerowe wiersze macierzy wyjść C, które zostały usunięte dla celów redukcji.</p>
transform2, transform_my_de c	<p>Parametry:</p> <p>path1 – ścieżka do folderu zawierającego zdekomponowane macierze stanu oraz macierze transformacji dla dekompozycji strukturalnej (transform2) lub Schur-</p>

	<p>Sylvester (<code>transform_my_dec</code>), <code>path2</code> – ścieżka do folderu, w którym zostanie zachowany model scalony, <code>n</code> – liczba podsystemów wydzielona w wyniku dekompozycji.</p> <p>Opis: Procedury scalają modele podsystemów wyznaczone w procesie dekompozycji strukturalnej lub Schur-Sylvester. Procedury zachowują scalony model w lokalizacji określonej parametrem <code>path2</code>.</p>
<code>dec_transform_schc</code>	<p>Parametry: <code>path</code> - ścieżka do folderu zawierającego zdekomponowane macierze stanu oraz macierze transformacji dla dekompozycji Schur-Complement, <code>n</code> - liczba modeli podsystemów wydzielonych w wyniku procesu dekompozycji.</p> <p>Opis: Procedura scala zredukowane modele podsystemów wyznaczone w procesie dekompozycji Schur-Complement.</p>
<code>schur_complement_connect_R3</code>	<p>Parametry: <code>path</code> – ścieżka do folderu zawierającego macierze stanu E,A,B,C modeli podsystemów, <code>path2</code> – ścieżka do folderu, w którym zostaną zapisane przekształcone macierze stanu.</p> <p>Opis: Procedura przekształca macierze uzyskane w wyniku dekompozycji Schur-Complement. Interfejs między modelami podsystemów, realizowany poprzez bloki macierzy E, przenoszony jest do macierzy A i B.</p>
<code>connect_sys</code>	<p>Parametry: <code>n</code> – liczba scalanych modeli, <code>path</code> – ścieżka do folderu zawierającego macierze stanu oraz macierze interfejsów scalanych modeli, <code>path2</code> – ścieżka do folderu, w którym zostaną zapisane macierze stanu modelu scalonego.</p> <p>Opis: Procedura scala <code>n</code> modeli podsystemów, których macierze stanu i interfejsy są zapisane w folderze określonym parametrem <code>path</code>. Wynikowy model zostaje zachowany w folderze określonym parametrem <code>path2</code>.</p>
<code>make_sys</code>	<p>Parametry: <code>path1</code> – ścieżka do folderu zawierającego macierze stanu modelu, <code>path2</code> – ścieżka do folderu, w którym zostanie zapisany obiekt ss.</p> <p>Opis: Procedura tworzy obiekt ss na podstawie macierzy A, B, C modelu i zachowuje go w folderze określonym parametrem <code>path2</code>.</p>
<code>transform_schc2</code>	<p>Parametry: <code>path1</code> – ścieżka do folderu zawierającego macierze stanu podsystemów modelu zdekomponowanego, <code>path2</code> – ścieżka do folderu, w którym zostaną zachowane przekształcone macierze stanu modeli podsystemów, <code>n</code> – liczba części, na które został podzielony model.</p> <p>Opis: Procedura wykorzystuje macierze transformacji w celu przekształcenia macierzy stanu E, A, B, C zdekomponowanych modeli podsystemów z wykorzystaniem dekompozycji Schur-Complement.</p>
<code>connect_sys4</code>	<p>Parametry: <code>n</code> – liczba scalanych modeli podsystemów, <code>path</code> – ścieżka do folderu zawierającego macierze stanu E, A, B, C scalanych modeli podsystemów, <code>path2</code> – ścieżka do folderu, w którym zostaną zapisane macierze stanu E, A, B, C modelu scalonego.</p> <p>Opis: Procedura łączy <code>n</code> modeli podsystemów, których macierze stanu są zapisane w folderze określonym parametrem <code>path</code>. Wynikowy model składający się z macierzy</p>

	stanu E, A, B, C zostaje zachowany w folderze określonym parametrem <code>path2</code> .
<code>make_sys_with_e</code>	<p>Parametry:</p> <p><code>path1</code> – ścieżka do folderu zawierającego macierze stanu E, A, B, C modelu, <code>path2</code> – ścieżka do folderu, w którym zostanie zapisany model w postaci obiektu ss.</p> <p>Opis:</p> <p>Procedura eliminuje macierz E modelu (poprzez przekształcenie równań stanu) i zachowuje wynikowy model w postaci obiektu ss.</p>
<code>give_sys_1</code>	<p>Parametry:</p> <p><code>a11</code>, <code>a12</code>, <code>a21</code>, <code>a22</code>, <code>e11</code>, <code>e12</code>, <code>e22</code>, <code>b1</code>, <code>b2</code> – macierze stanu podsystemów zdekomponowanych i zredukowanych.</p> <p>Wyniki:</p> <p><code>a11</code> – macierz stanu A podsystemu pierwszego, <code>a12</code> – interfejs określający wpływ zmiennych stanu podsystemu drugiego na zmienne stanu podsystemu pierwszego, <code>b1</code> – macierz wejść podsystemu pierwszego.</p> <p>Opis:</p> <p>Funkcja wyznacza macierze stanu A,B,C podsystemu pierwszego na podstawie zdekomponowanych metodą Schur-Complement i zredukowanych macierzy stanu podsystemów.</p>
<code>give_sys_2</code>	<p>Parametry:</p> <p><code>a11</code>, <code>a12</code>, <code>a21</code>, <code>a22</code>, <code>e11</code>, <code>e12</code>, <code>e22</code>, <code>b1</code>, <code>b2</code> – macierze stanu podsystemów zdekomponowanych i zredukowanych,</p> <p>Wyniki:</p> <p><code>a22</code> – macierz stanu A podsystemu drugiego, <code>a21</code> – interfejs określający wpływ zmiennych stanu podsystemu pierwszego na zmienne stanu podsystemu drugiego, <code>b2</code> – macierz wejść podsystemu drugiego</p> <p>Opis:</p> <p>Funkcja wyznacza macierze stanu A,B,C podsystemu drugiego na podstawie zdekomponowanych metodą Schur-Complement i zredukowanych macierzy stanu podsystemów.</p>
<code>place_matrix</code>	<p>Parametry:</p> <p><code>vec</code> – wektor zawierający rząd macierzy stanu A podsystemów zdekomponowanych,</p> <p>Wyniki:</p> <p><code>x1</code>, <code>x2</code> – współrzędne wierszowe lub kolumnowe określające umiejscowienie macierzy stanu podsystemu dodawanego do modelu scalonego.</p> <p>Opis:</p> <p>Funkcja na podstawie wektora zawierającego rozmiar dodawanych do modelu scalanego podsystemów, wyznacza współrzędne wierszowe lub kolumnowe określające umiejscowienie bloku związanego z kolejnym podsystemem dla procesu konkatencji macierzy.</p>
<code>define_ff</code> , <code>define_fb</code>	<p>Parametry:</p> <p><code>n</code> – wektor zawierający rozmiar macierzy stanu A modeli zdekomponowanych, <code>i</code> – indeks pierwszego podsystemu, <code>j</code> – indeks drugiego podsystemu,</p> <p>Wyniki:</p> <p><code>x1</code>, <code>y1</code>, <code>x2</code>, <code>y2</code> – współrzędne wierszowe (<code>x1</code>, <code>x2</code>) oraz kolumnowe (<code>y1</code>, <code>y2</code>) określające lokalizację interfejsu $\Gamma_{i,j}$ (dla funkcji <code>define_ff</code>) lub $\Gamma_{j,i}$ (dla funkcji <code>define_fb</code>) w macierzy stanu A modelu zdekomponowanego.</p> <p>Opis:</p> <p>Funkcje określają umiejscowienie bloków związanych z interfejsami pomiędzy podsystemami zdekomponowanymi w macierzy stanu A modelu pierwotnego.</p>
<code>str_con_zm2</code>	<p>Parametry:</p> <p><code>path</code> – ścieżka do folderu zawierającego macierze stanu i interfejsy pomiędzy modelami podsystemów uzyskanymi w wyniku dekompozycji strukturalnej uproszczonej, <code>n</code> – liczba wydzielonych podsystemów,</p> <p>Opis:</p>

	Procedura wykorzystuje procedury <code>transform_n</code> oraz <code>str_upr_connect</code> w celu wyznaczenia modelu scalonego, dla przypadku dekompozycji strukturalnej uproszczonej.
<code>transform_n</code>	<p>Parametry:</p> <p><code>path1</code> – ścieżka do folderu zawierającego macierze stanu i interfejsy pomiędzy modelami podsystemów, uzyskanymi w wyniku dekompozycji strukturalnej uproszczonej,</p> <p><code>path2</code> – ścieżka do folderu, w którym zostaną zapisane przekształcone macierze stanu modeli podsystemów i interfejsy,</p> <p><code>n</code> – liczba wydzielonych podsystemów.</p> <p>Opis:</p> <p>Procedura realizuje przekształcenie modeli podsystemów i interfejsów, uzyskanych w wyniku dekompozycji strukturalnej uproszczonej, z wykorzystaniem macierzy transformacji redukcji.</p>
<code>str_upr_connect</code>	<p>Parametry:</p> <p><code>path1</code> – ścieżka do zredukowanych modeli podsystemów i interfejsów wyznaczonych w wyniku dekompozycji strukturalnej uproszczonej,</p> <p><code>path2</code> – ścieżka do folderu, w którym zostanie zapisany model scalony,</p> <p><code>n</code> – liczba wydzielonych podsystemów.</p> <p>Opis:</p> <p>Procedura wyznacza model scalony na podstawie zredukowanych modeli i interfejsów uzyskanych w wyniku dekompozycji strukturalnej uproszczonej.</p>
<code>dec_transform_s</code> <code>chc_zm2</code>	<p>Parametry:</p> <p><code>path</code> – ścieżka do folderu zawierającego macierze stanu A,B,C zdekomponowanych modeli podsystemów.</p> <p>Opis:</p> <p>Procedura przekształca macierze uzyskane w wyniku zmodyfikowanej dekompozycji Schur complement z wykorzystaniem wyznaczonych macierzy transformacji redukcji.</p>
<code>connect_sys_sch</code> <code>c_zm,</code> <code>transform2_zm,</code> <code>transform_my_de</code> <code>c_zm,</code>	<p>Parametry:</p> <p><code>n</code> – liczba wydzielonych podsystemów,</p> <p><code>path</code> – ścieżka do folderu zawierającego zredukowane macierze stanu podsystemów i interfejsy wyznaczone w procesie dekompozycji,</p> <p><code>path2</code> – ścieżka do folderu, w którym zostanie zapisany model scalony.</p> <p>Opis:</p> <p>Procedury scalają zredukowane modele podsystemów dla przypadku dekompozycji: Schur complement zmodyfikowanej, strukturalnej, Schur Complement. Wynikowy model zapisany zostaje do folderu określonego parametrem <code>path2</code>.</p>
<code>dec_transform_s</code> <code>chc</code>	<p>Parametry:</p> <p><code>path</code> - ścieżka do folderu zawierającego macierze stanu podsystemów oraz macierze transformacji redukcji,</p> <p><code>n</code> - liczba wydzielonych podsystemów.</p> <p>Opis:</p> <p>Procedura łączy zredukowane podsystemy dla przypadku dekompozycji Schur complement.</p>
<code>schur_complemen</code> <code>t_connect_R3_zm</code>	<p>Parametry:</p> <p><code>path</code> – ścieżka do folderu zawierającego macierze stanu E,A,B,C zdekomponowanych modeli podsystemów,</p> <p><code>path2</code> – ścieżka do folderu, w którym zostaną zapisane przekształcone macierze stanu A,B,C modeli zdekomponowanych.</p> <p>Opis:</p> <p>Procedura przekształca macierze uzyskane w wyniku dekompozycji Schur complement dwupoziomowej. Interfejs między modelami realizowany poprzez bloki macierzy E przenoszony jest do macierzy A i B.</p>
<code>transform_schc_</code> <code>zm</code>	<p>Parametry:</p> <p><code>path1</code> – ścieżka do folderu zawierającego macierze stanu podsystemów,</p> <p><code>path2</code> – ścieżka do folderu, w którym zostaną zachowane przekształcone macierze stanu,</p> <p><code>n</code> – liczba wydzielonych podsystemów.</p> <p>Opis:</p>

	Procedura wykorzystuje macierze transformacji redukcji w celu przekształcenia macierzy stanu E zdekomponowanych z wykorzystaniem dekompozycji Schur Complement (dwupoziomowej) modeli podsystemów.
fol_of_sys	Parametry: type – rodzaj dekompozycji. Wyniki: fol – nazwa folderu. Opis: Funkcja na podstawie typu dekompozycji określonego liczbą przekazaną w parametrze type zwraca nazwę folderu, w którym zostaną zachowane pliki związane z daną dekompozycją.
dist_str_full_reduct, dist_my_dec_reduct, dist_mls_reduct, dist_str_upr_reduct, dist_schur_dec_full_reduct	Parametry: path – ścieżka do folderu zawierającego zdekomponowane modele podsystemów, n – liczba wydzielonych podsystemów, om_min – dolny zakres aproksymacji charakterystyk częstotliwościowych modelu zredukowanego, om_max – górny zakres aproksymacji charakterystyk częstotliwościowych modelu zredukowanego k – docelowy rozmiar modelu zredukowanego. Opis: Zbiór procedur realizujących równoległą redukcję zdekomponowanego modelu dla przypadku dekompozycji jednopoziomowej nie związanej z dekompozycją slow-fast.
dist_slow_fast_reduct, dist_slow_fast_reduct2, dist_slow_fast_reduct3, dist_slow_fast_reduct11, dist_slow_fast_reduct14	Parametry: path – ścieżka do folderu zawierającego zdekomponowane modele podsystemów, n – liczba wydzielonych podsystemów, om_min – dolny zakres aproksymacji charakterystyk częstotliwościowych części wolnej modelu zredukowanego, om_max – górny zakres aproksymacji charakterystyk częstotliwościowych części wolnej modelu zredukowanego, k – docelowy rozmiar części wolnej modelu zredukowanego, om_min_fast – dolny zakres aproksymacji charakterystyk częstotliwościowych części szybkiej modelu zredukowanego, om_max_fast – górny zakres aproksymacji charakterystyk częstotliwościowych części szybkiej modelu zredukowanego k_fast – docelowy rozmiar części szybkiej modelu zredukowanego. Opis: Zbiór procedur realizujących równoległą redukcję zdekomponowanego modelu dla przypadku dekompozycji jednopoziomowej slow-fast.
eps_str_con, eps_stru_con, eps_my_dec_con, eps_schc_con, eps_schc_zm_con, eps_sch_con, eps_MATLAB_con	Parametry: path – ścieżka do folderu zawierającego macierze stanu i interfejsy pomiędzy modelami podsystemów uzyskanymi w wyniku dekompozycji <i>Esf</i> , Opis: Procedury scalają zredukowaną część wolną i część szybką dla przypadku dekompozycji: <i>Esf-Str</i> , <i>Esf-StrUpr</i> , <i>Esf-SchSylv</i> , <i>Esf-SchComp</i> , <i>Esf-SchCompZm</i> , <i>Esf-Sch</i> , MATLAB slowfast.
dist_slow_fast_reduct_fast, dist_slow_fast_reduct2_fast, dist_slow_fast_reduct3_fast, dist_slow_fast_reduct11_fast, dist_slow_fast_reduct14_fast	Parametry: path – ścieżka do folderu zawierającego zdekomponowane modele podsystemów, n – liczba wydzielonych podsystemów, om_min_fast – dolny zakres aproksymacji charakterystyk częstotliwościowych części szybkiej modelu zredukowanego, om_max_fast – górny zakres aproksymacji charakterystyk częstotliwościowych części szybkiej modelu zredukowanego, k_fast – docelowy rozmiar części szybkiej modelu zredukowanego. Opis: Zbiór procedur realizujących redukcję części szybkiej zdekomponowanego modelu dla przypadku dekompozycji wielopoziomowej, gdzie na pierwszym poziomie została wykonana dekompozycja slow-fast.

dist4	<p>Parametry:</p> <p><code>paths_models</code> – tablica komórek (ang. cell array) zawierająca w kolejnych wierszach ścieżki do folderów, w których zachowane są macierze stanu zredukowanych modeli,</p> <p><code>paths_results</code> – tablica komórek (ang. cell array) zawierająca w kolejnych wierszach ścieżki do folderów, w których zostaną zachowane modele wynikowe,</p> <p><code>types</code> – macierz zawierająca informacje na temat typów dekompozycji realizowanych na poszczególnych poziomach dekompozycji dla zredukowanych modeli. Jeden wiersz macierzy zawiera komplet danych opisujących typy dekompozycji jednego modelu. Każda kolumna macierzy opisuje inny poziom dekompozycji.</p> <p><code>ns</code> – macierz zawierająca informacje na temat liczby wydzielanych części na konkretnym poziomie dekompozycji dla zredukowanych modeli. Jeden wiersz macierzy zawiera komplet danych opisujących liczbę wydzielonych części na konkretnym poziomie dekompozycji dla jednego zredukowanego modelu.</p> <p><code>om_mins</code> - macierz zawierająca informacje na temat dolnego zakresu aproksymacji charakterystyk częstotliwościowych modeli zredukowanych (w przypadku realizacji dekompozycji slow-fast macierz zawiera informacje na temat dolnego zakresu aproksymacji charakterystyk częstotliwościowych części wolnej modeli). Każdy wiersz macierzy zawiera komplet informacji dla danego modelu.</p> <p><code>om_maxs</code> - macierz zawierająca informacje na temat górnego zakresu aproksymacji charakterystyk częstotliwościowych modeli zredukowanych (w przypadku realizacji dekompozycji slow-fast macierz zawiera informacje na temat górnego zakresu adekwatności części wolnej modeli). Każdy wiersz macierzy zawiera komplet informacji dla danego modelu.</p> <p><code>ks</code> – wektor zawiera docelowe rozmiary zredukowanych modeli (w przypadku realizacji dekompozycji slow-fast wektor określa rozmiar części wolnej modelu zredukowanego). Każdy wiersz wektora określa rozmiar modelu zredukowanego dla konkretnego modelu,</p> <p><code>om_min_fasts</code> - macierz zawierająca informacje na temat dolnego zakresu aproksymacji charakterystyk częstotliwościowych części szybkiej modeli zredukowanych. Każdy wiersz macierzy zawiera komplet informacji dla danego modelu.</p> <p><code>om_max_fasts</code> - macierz zawierająca informacje na temat górnego zakresu aproksymacji charakterystyk częstotliwościowych części szybkiej modeli zredukowanych. Każdy wiersz macierzy zawiera komplet informacji dla danego modelu.</p> <p><code>k_fasts</code> - wektor zawiera docelowe rozmiary części szybkiej zredukowanych modeli. Każdy wiersz wektora określa rozmiar części szybkiej modelu zredukowanego dla konkretnego modelu.</p> <p>Opis:</p> <p>Procedura realizuje równoległą redukcję wielu modeli. Operacje wykonywane wewnątrz procedury są następujące:</p> <ol style="list-style-type: none"> 1. Wyszukuje menedżera zadań na komputerze określonym daną nazwą 2. Określa czy menedżer zadań dysponuje przynajmniej dwiema wolnymi (ang. idle) jednostkami roboczymi (ang. workers), ze względu na fakt, że dodanie kolejnego zadania redukcji powiązanej z dekompozycją wymaga uruchomienia go przynajmniej na dwóch jednostkach roboczych. W przypadku, gdy wszystkie jednostki robocze wykonują w danej chwili obliczenia, procedura wraca do kroku 1. 3. Wyznacza liczbę potrzebnych jednostek roboczych (ang. workers) w celu realizacji kolejnego zadania redukcji (<code>computational_resources</code>). 3.1. W przypadku, gdy liczba wolnych jednostek roboczych jest mniejsza od wymagań określonych w punkcie 3, procedura cyklicznie sprawdza stan uruchomionych zadań (ang. jobs) w poszukiwaniu zadania ukończonego. 3.2. Po odszukaniu zadania ukończonego, wyniki jego działania (macierze modelu zredukowanego, kod ewentualnego błędu) są pobierane i zapisywane do właściwego katalogu stworzonej hierarchii plików. Ponownie wykonywany jest krok 3.1 4. W przypadku dostępności wystarczającej liczby jednostek obliczeniowych do realizacji kolejnego zadania redukcji, tworzone jest nowe zadanie (ang. job)
-------	---

	<p>5. Określana jest zależność folderów pomiędzy poszczególnymi węzłami klastra,</p> <p>6. Dodawany jest wątek (ang. task) związany z dekompozycją oraz redukcją dla kolejnego modelu. Odbywa się to poprzez zdalne wywołanie funkcji <code>dist_computation</code>.</p> <p>7. Utworzone zadanie (ang. job) dodawane jest do kolejki zadań. Procedura wykonuje ponownie krok 1, aż do wyczerpania modeli określonych parametrem <code>paths_models</code></p>
<code>computational_resources</code>	<p>Parametry:</p> <p><code>ns</code> – wektor zawierający dane na temat liczby wydzielanych części na każdym z poziomów dekompozycji modelu</p> <p>Wyniki:</p> <p><code>n</code> – liczba jednostek roboczych (ang. workers) konieczna do realizacji zadania (ang. job)</p> <p>Opis:</p> <p>Procedura oblicza liczbę potrzebnych jednostek roboczych dla wykonania zadania. Liczba ta jest równa liczbie wydzielonych modeli na ostatnim poziomie dekompozycji (dla każdego modelu zostanie wywołana osobna realizacja redukcji FW) plus jeden (jedna jednostka robocza związana z dekompozycją i scalaniem modelu)</p>
<code>dist_computation</code>	<p>Parametry:</p> <p><code>a, b, c</code> – macierze stanu modelu oryginalnego,</p> <p><code>type</code> – wektor zawierający typ dekompozycji dla poszczególnych poziomów</p> <p><code>n</code> – wektor określający na ile podsystemów należy zdekomponować model na kolejnych poziomach,</p> <p><code>om_min</code> - dolny zakres aproksymacji charakterystyk częstotliwościowych modelu zredukowanego (w przypadku dekompozycji slow-fast parametr określa dolny zakres aproksymacji charakterystyk częstotliwościowych części wolnej modelu zredukowanego),</p> <p><code>om_max</code> – górny zakres aproksymacji charakterystyk częstotliwościowych modelu zredukowanego (w przypadku dekompozycji slow-fast parametr określa górny zakres aproksymacji charakterystyk częstotliwościowych dla części wolnej modelu zredukowanego),</p> <p><code>k</code> – docelowy rozmiar modelu zredukowanego (w przypadku dekompozycji slow-fast rozmiar części wolnej modelu zredukowanego),</p> <p><code>om_min_fast</code> – dolny zakres aproksymacji charakterystyk częstotliwościowych części szybkiej modelu zredukowanego,</p> <p><code>om_max_fast</code> – górny zakres aproksymacji charakterystyk częstotliwościowych części szybkiej modelu zredukowanego,</p> <p><code>k_fast</code> – docelowy rozmiar części szybkiej modelu zredukowanego.</p> <p>Wyniki:</p> <p><code>sys</code> – macierze stanu modelu zredukowanego,</p> <p><code>blad</code> – kod ewentualnego błędu (związany z uzyskaniem modelu niestabilnego na dowolnym poziomie dekompozycji, zerowych macierzy stanu B lub C).</p> <p>Opis:</p> <p>Funkcja uruchamiana na jednostkach roboczych klastra MATLAB wywołuje funkcję <code>dist_multi_dec_fin</code> w celu dekompozycji i redukcji pojedynczego modelu.</p>
<code>primary_read</code>	<p>Parametry:</p> <p><code>path</code> – ścieżka do pliku tekstowego.</p> <p>Wyniki:</p> <p><code>tekst</code> – tablica znaków char zawierająca zawartość odczytanego pliku.</p> <p>Opis:</p> <p>Funkcja odczytuje plik tekstowy zawarty w bieżącej lokalizacji i zwraca jego zawartość. Została zaimplementowana w celu określenia zależności pomiędzy plikami i folderami dla wszystkich jednostek roboczych (ta sama zawartość zmiennej środowiskowej <code>path</code>).</p>
<code>del_dist</code>	<p>Opis:</p> <p>Procedura usuwa wszystkie elementy zawarte w folderze <code>C:\dist</code> na wszystkich węzłach klastra. Folder ten zawiera pliki tymczasowe wykorzystywane przez proces równoległej redukcji.</p>

DODATEK C. Wybrane funkcje środowiska MATLAB związane z wyznaczaniem rozwiązania równania Lapunowa

Parametry:

A, B – macierze stanu (A,B dla gramianu obserwowalności, A', C' dla gramianu sterowalności).

Wyniki:

gramian - gramian sterowalności lub obserwowalności.

Wybrane funkcje środowiska MATLAB związane z wyznaczeniem rozwiązania równania Lapunowa	
Nazwa	Opis
smi, smi2	Funkcje wyznaczają gramian sterowalności lub obserwowalności metodą Smith'a. Funkcja smi realizuje z góry narzuconą liczbę iteracji, smi2 realizuje algorytm do momentu, gdy różnica wyników z poprzedniej i bieżącej iteracji spełnia przyjętą spełnia warunek zatrzymania (maksymalny element macierzy mniejszy od 10^{-8}).
sfn, sfn2_c	Funkcje wyznaczają gramian sterowalności lub obserwowalności metodą funkcji znaku.
bartels	Funkcja wyznacza gramian sterowalności lub obserwowalności metodą Bartels-Stewart'a z wykorzystaniem zespolonej dekompozycji Schur'a.
bartels_real	Funkcja wyznacza gramian sterowalności lub obserwowalności metodą Bartels-Stewart'a z wykorzystaniem rzeczywistej dekompozycji Schur'a.
ADI, ADI2	Funkcje wyznaczają gramian sterowalności lub obserwowalności metodą ADI (dla określonej liczby iteracji lub różnicy wyników)
LYAP_KRON	Funkcja wyznacza gramian sterowalności lub obserwowalności metodą Kroneckera.

DODATEK D. Wybrane funkcje i procedury języka C++ związane z realizacją obliczeń z wykorzystaniem procesora graficznego

Wybrane funkcje i procedury języka C++ związane z realizacją obliczeń na karcie graficznej	
Nazwa	Opis
to_card	Kopiuje macierz z pamięci głównej do pamięci karty graficznej
From_card	Kopiuje macierz z pamięci karty graficznej do pamięci głównej
del_dev	Usuwa macierz z pamięci karty graficznej
del_host	Usuwa macierz z pamięci głównej
Operator *	Iloczyn macierzy
mul1	Wykonanie operacji $\mathbf{X} = \alpha\mathbf{AB}$ z wykorzystaniem procedury gemm z biblioteki CULA
mul2	Wykonanie operacji $\mathbf{X} = \alpha\mathbf{AB}^*$ z wykorzystaniem procedury gemm z biblioteki CULA
Operator +	Suma macierzy
add_in_place	Suma macierzy, wynik zapisywany jest w obiekcie, z którego została wywołana metoda
Operator -	Różnica macierzy
Operator !	Konwersja macierzy rzeczywistej na zespoloną lub zespolonej na rzeczywistą
sub_in_place	Różnica macierzy, wynik zapisywany jest w obiekcie, z którego została wywołana metoda
Operator *	Mnożenie macierzy przez skalar
Operator ++	Zwraca macierz transponowaną

(int)	
Operator ++	Transponuje macierz, nadpisując dane źródłowe
Dup	Zwraca kopię macierzy
Inv	Inwersja macierzy
Operator ^	Potęga macierzy (również potęgi ujemne)
Operator / (CulyaMatrix)	Rozwiązanie układu równań $AX=B$, gdzie A jest lewym operandem a B prawym. Wykorzystuje procedurę <code>gesv</code>
Operator /=	Rozwiązanie układu równań $AX=B$, gdzie A macierz trójkątna. Wykorzystuje procedurę <code>CulaDeviceZtrtrs</code>
Operator %	Rozwiązanie układu równań $AX=B$. Wykorzystuje iteracyjną wersję procedury <code>gesv</code>
Operator >>	Wypełnia macierz podaną jako parametr (prawa strona) częścią macierzy stanowiącej drugi parametr (lewa strona).
Qr	Wyznaczenie macierzy R z dekompozycji QR. Metoda wykorzystuje procedurę <code>geqrf</code> z biblioteki CULA. Wynikiem działania jest trójkątna górna macierz R
tau_r_qr	Wyznaczenie macierzy R oraz wektora tau dla dekompozycji QR. Metoda wykorzystuje procedurę <code>geqrf</code> z biblioteki CULA. Wynikiem działania jest trójkątna górna macierz R oraz macierz tau , wykorzystywana przy wyznaczaniu macierzy Q
q_qr	Wyznaczenie macierzy Q z dekompozycji QR. Metoda wykorzystuje procedury <code>ungqr</code> oraz <code>orgqr</code> z biblioteki CULA.
qr_full	Dekompozycja QR macierzy. Metoda zwraca obiekt klasy TQr zawierający macierz R oraz Q z dekompozycji
eig_full	Wyznaczenie wartości i wektorów własnych macierzy. Metoda zwraca obiekt klasy TEig zawierający wektor wartości własnych oraz macierz prawych i lewych wektorów własnych
svd	Dekompozycja SVD macierzy. Metoda zwraca obiekt klasy TSvd zawierający wektor wartości szczególnych oraz macierze U i V *
Cholesky	Dekompozycja Cholesky'ego macierzy
cholesky2	Dekompozycja Cholesky'ego macierzy. Metoda w przypadku wystąpienia błędu przy wyznaczaniu dekompozycji Cholesky'ego zwraca wartość true
my_schur	Wyznaczenie postaci Schur'a macierzy
my_schur_full	Dekompozycja Schur'a macierzy. Metoda zwraca obiekt klasy TSchur zawierający macierz postaci Schur'a oraz unitarną macierz transformacji
schur_123	Dekompozycja Schur'a macierzy. Metoda zwraca obiekt klasy TSchur zawierający macierz postaci Schur'a, w której elementy na przekątnej uporządkowane są rosnąco oraz unitarną macierz transformacji
schur_321	Dekompozycja Schur'a macierzy. Metoda zwraca obiekt klasy TSchur zawierający macierz postaci Schur'a, w której elementy na przekątnej uporządkowane są malejąco oraz unitarną macierz transformacji
schur_real_123	Dekompozycja Schur'a macierzy. Metoda zwraca obiekt klasy TSchur zawierający macierz postaci Schur'a, w której elementy na przekątnej uporządkowane są rosnąco względem ich części rzeczywistej oraz macierz transformacji.
schur_real_321	Dekompozycja Schur'a macierzy. Metoda zwraca obiekt klasy TSchur zawierający macierz postaci Schur'a, w której elementy na przekątnej uporządkowane są malejąco względem ich części rzeczywistej oraz macierz transformacji
full_spectrum	Wyznaczenie wartości własnych macierzy. Metoda zwraca obiekt klasy TSpectrum, który w zależności od typu macierzy zwraca wektor zespolony lub rzeczywisty z wartościami własnymi
make_up	Wyzerowanie elementów poniżej przekątnej macierzy
X_plus_a_mul_Y	Wykonanie operacji $X = X + a_i Y$
X_plus_a_mul_Y	Wykonanie operacji $X = X + a_{i,j} Y$
givens2	Wyznaczenie zespolonej macierzy rotacji Givens'a $X = \begin{bmatrix} c & s \\ -s^* & c \end{bmatrix} \quad \text{tak} \quad \text{aby} \quad X \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}, \quad \text{gdzie}$

	$c = \frac{ a_1 }{\sqrt{\operatorname{Re}(a_1)^2 + \operatorname{Im}(a_1)^2 + \operatorname{Re}(a_2)^2 + \operatorname{Im}(a_2)^2}}$ $s = \frac{a_1}{\sqrt{\operatorname{Re}(a_1)^2 + \operatorname{Im}(a_1)^2}} \cdot \frac{a_2^*}{\sqrt{\operatorname{Re}(a_1)^2 + \operatorname{Im}(a_1)^2 + \operatorname{Re}(a_2)^2 + \operatorname{Im}(a_2)^2}}$ <p>gdzie a_1 jest elementem macierzy o współrzędnych x_1, y_1, a_2 jest elementem macierzy o współrzędnych x_2, y_2 (parametry)</p>
givens2_1	<p>Wyznaczenie zespolonej macierzy rotacji Givens'a</p> $X = \begin{bmatrix} c & s \\ -s^* & c \end{bmatrix}$ <p>tak aby $X \begin{bmatrix} a_1 \\ a_2 - a_3 \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}$, gdzie a_3 jest elementem o współrzędnych x_{2+1}, y_{2+1} (parametr)</p>
givens_cycle	<p>Wykonanie rotacji Givensa w postaci $X \begin{bmatrix} a_1 \\ a_2 - a_3 \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}$ dla k i $k+1$ wiersza macierzy.</p>
givens	<p>Wyznaczenie uporządkowanej postaci Schura macierzy, gdzie elementy na przekątnej ułożone są w porządku rosnącym. Jednocześnie wyznaczana jest macierz \mathbf{q}, taka, że $\mathbf{A} = \mathbf{qtq}^*$. Metoda jako parametry pobiera nieuporządkowaną postać Schur'a macierzy (\mathbf{t}) oraz unitarną macierz transformacji (\mathbf{q}).</p>
givens_real	<p>Wyznaczenie uporządkowanej postaci Schura macierzy, gdzie elementy na przekątnej ułożone są w porządku rosnącym. Metoda działa poprawnie dla macierzy \mathbf{t} i \mathbf{q} rzeczywistych</p>
givens_real2	<p>Wyznaczenie uporządkowanej postaci Schura macierzy, gdzie elementy na przekątnej ułożone są w porządku malejącym. Metoda działa poprawnie dla macierzy \mathbf{t} i \mathbf{q} rzeczywistych</p>

DODATEK E. Wybrane funkcje i procedury języka C++ związane z wyznaczeniem rozwiązania równania Lapunowa w środowisku procesora graficznego

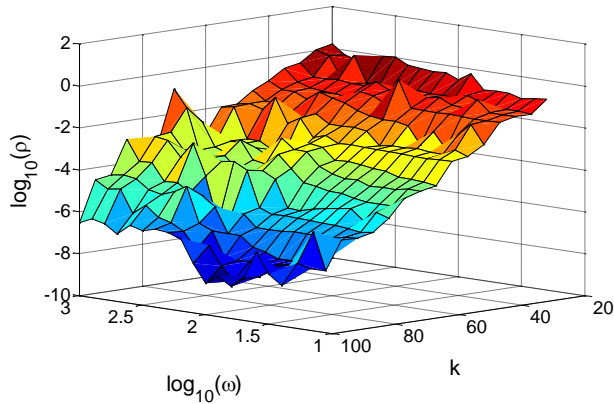
Wybrane funkcje i procedury języka C++ związane z wyznaczeniem rozwiązania równania Lapunowa w środowisku karty graficznej	
Nazwa	Opis
del_system	Usunięcie z pamięci głównej macierzy stanu A, B, C zredukowanego modelu
del_all	Usunięcie z pamięci głównej oraz pamięci karty graficznej wszystkich załadowanych macierzy
Smith	Wyznaczenie gramianów sterowalności i obserwowalności modelu metodą Smith'a
Sfn	Wyznaczenie gramianów sterowalności i obserwowalności modelu metodą funkcji znaku
Adi	Wyznaczenie gramianów sterowalności i obserwowalności modelu metodą ADI
Bs	Wyznaczenie gramianów sterowalności i obserwowalności modelu metodą Bartels-Stewart'a
Chol_P	Dekompozycja Cholesky'ego gramianu sterowalności
Chol_Q	dekompozycja Cholesky'ego gramianu obserwowalności
Chol_Q2	Dekompozycja Cholesky'ego gramianu obserwowalności. W przypadku, gdy gramian obserwowalności nie jest dodatnio określony (w wyniku błędów numerycznych) metoda dodaje do elementów leżących na diagonalu niewielką wartość aż do momentu, gdy gramian będzie dodatnio określony
Chol_P2	Dekompozycja Cholesky'ego gramianu sterowalności. W przypadku, gdy gramian sterowalności nie jest dodatnio określony (w wyniku błędów numerycznych) metoda dodaje do elementów leżących na diagonalu niewielką wartość aż do momentu, gdy gramian będzie dodatnio określony
Smith	Rozwiązanie równania Lapunowa metodą Smith'a
smith2	Rozwiązanie równania Lapunowa metodą Smith'a. Metoda kończy działanie, gdy różnica elementów macierzy uzyskana w kolejnych iteracjach jest mniejsza od wartości przekazanej w parametrze (brana jest pod uwagę wartość bezwzględna największego elementu z macierzy różnicy)
cf_smith	Wyznaczenie dekompozycji Cholesky'ego rozwiązania równania Lapunowa metodą Smith'a
std_sign_f	Rozwiązanie równania Lapunowa metodą funkcji znaku
std_sign_f2	Rozwiązanie równania Lapunowa metodą funkcji znaku. Metoda kończy działanie, gdy różnica elementów macierzy uzyskana w kolejnych iteracjach jest mniejsza od wartości przekazanej w parametrze (brana jest pod uwagę wartość bezwzględna największego elementu z macierzy różnicy)
std_cf_sign_f	Wyznaczenie dekompozycji Cholesky'ego rozwiązania równania Lapunowa metodą funkcji znaku
Bartels5	Rozwiązanie równania Lapunowa metodą Bartels-Stewart'a
ADI_STD2	Rozwiązanie równania Lapunowa metodą ADI. Metoda wykonuje liczbę iteracji algorytmu określoną parametrem
ADI_STD3	Rozwiązanie równania Lapunowa metodą ADI. Metoda kończy działanie, gdy różnica elementów macierzy uzyskana w kolejnych iteracjach jest mniejsza od wartości przekazanej w parametrze (brana jest pod uwagę wartość bezwzględna największego elementu z macierzy różnicy)
shifts	Wyznaczenie wektora przesunięć dla kolejnych iteracji algorytmu ADI metodą podaną przez T. Penzl'a
shifts2	Wyznaczenie określonej liczby przesunięć dla kolejnych iteracji algorytmu ADI na podstawie parametrów w postaci wartości własnych oraz wektora wcześniej wyznaczonych przesunięć

DODATEK F. Wybrane funkcje i procedury języka C++ związane z redukcją SVD w środowisku procesora graficznego

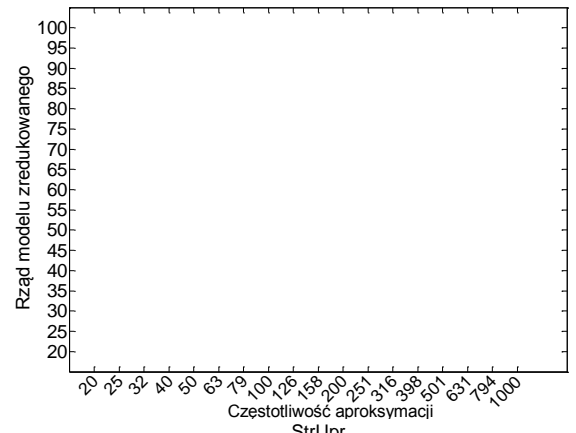
Wybrane funkcje i procedury języka C++ związane z redukcją SVD w środowisku karty graficznej	
Nazwa	Opis
rpr	Wyznaczenie macierzy transformacji metodą RPR. Parametr funkcji określa rząd modelu zredukowanego
sr	Wyznaczenie macierzy transformacji metodą SR. Parametr funkcji określa rząd modelu zredukowanego
bf	Wyznaczenie macierzy transformacji metodą Schur'a. Parametr funkcji określa rząd modelu zredukowanego
Bfsr_a	Wyznaczenie macierzy transformacji metodą BFSR wersja 1. Parametr funkcji określa rząd modelu zredukowanego
Bfsr_b	Wyznaczenie macierzy transformacji metodą BFSR wersja 2. Parametr funkcji określa rząd modelu zredukowanego
Obi	Wyznaczenie macierzy transformacji metodą Obinata-Anderson'a. Parametr funkcji określa rząd modelu zredukowanego
Eigbfsrb	Wyznaczenie macierzy transformacji metodą EIG-BFSR wersja 2. Parametr funkcji określa rząd modelu zredukowanego
Eigbfsra	Wyznaczenie macierzy transformacji metodą EIG-BFSR wersja 1. Parametr funkcji określa rząd modelu zredukowanego
Eigsr	Wyznaczenie macierzy transformacji metodą EIG-SR. Parametr funkcji określa rząd modelu zredukowanego
Bta	Redukcja wymiaru modelu z wykorzystaniem metody BTA

Dodatek G. – wykresy błędów i mapy stabilności dla redukcji powiązanej z dekompozycją jednopoziomową

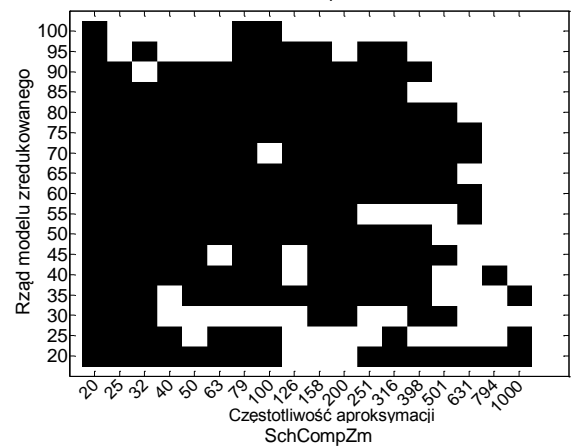
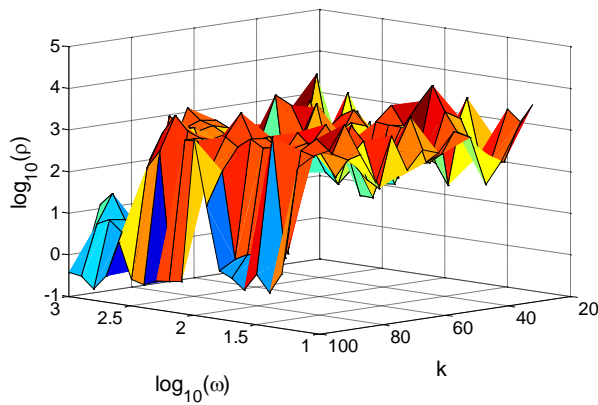
Błąd redukcji:
SchSylv



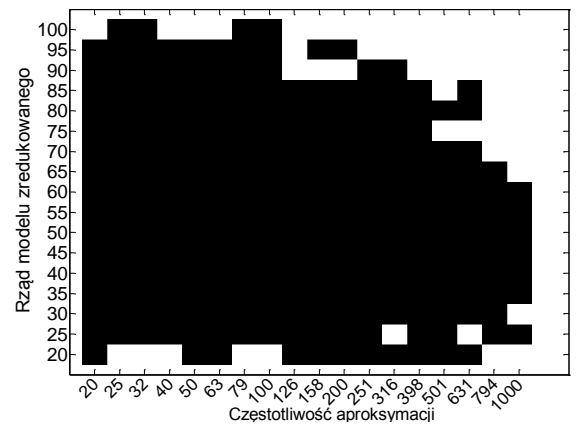
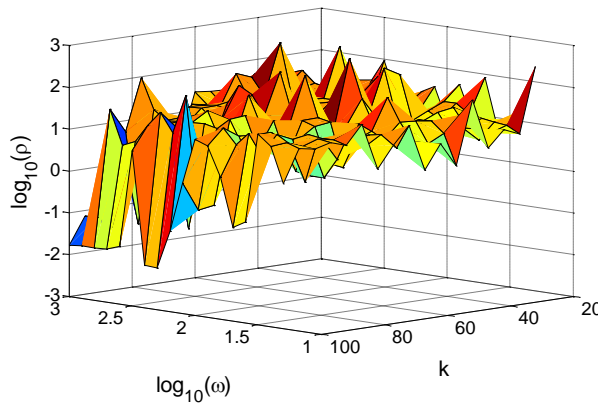
Mapa stabilności:
SchSylv

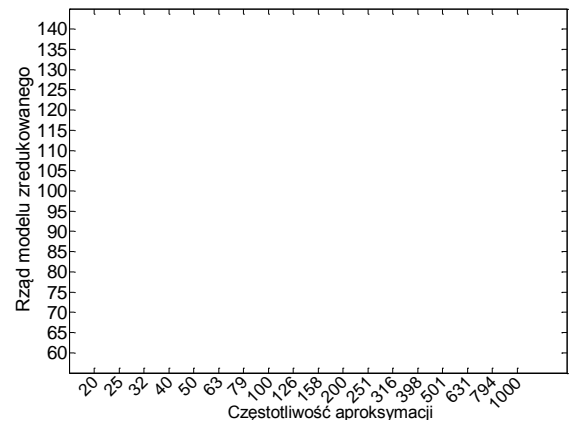
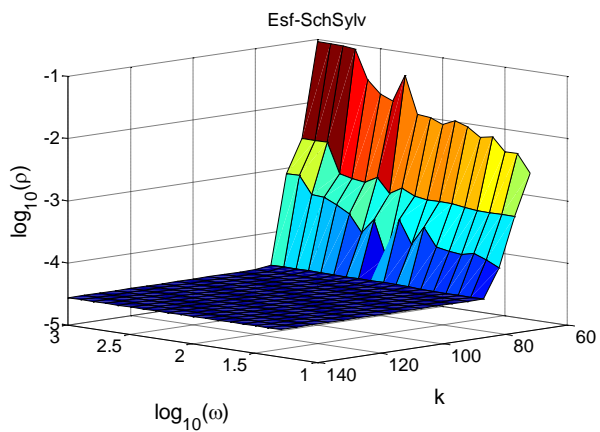
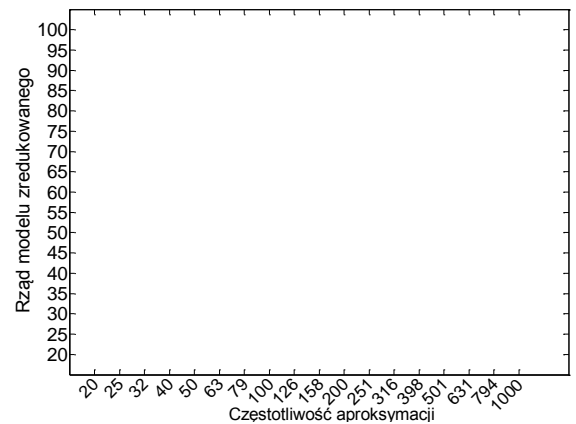
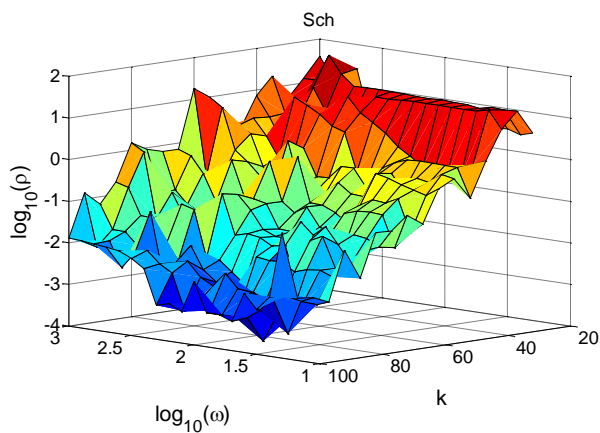
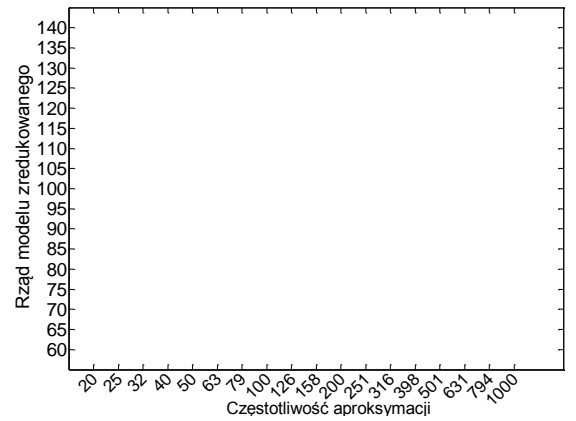
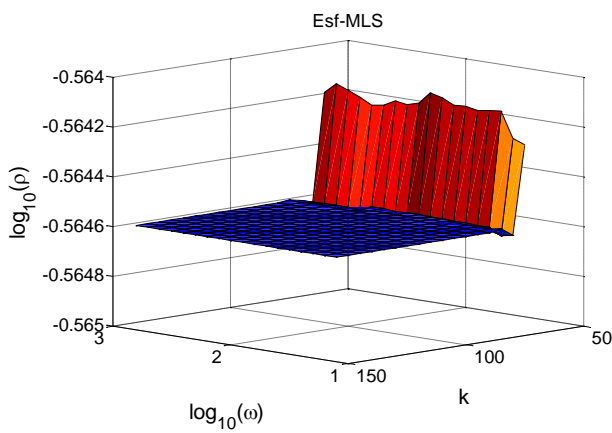
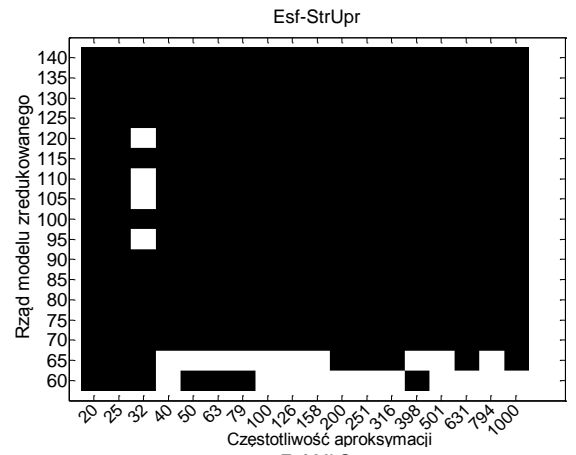
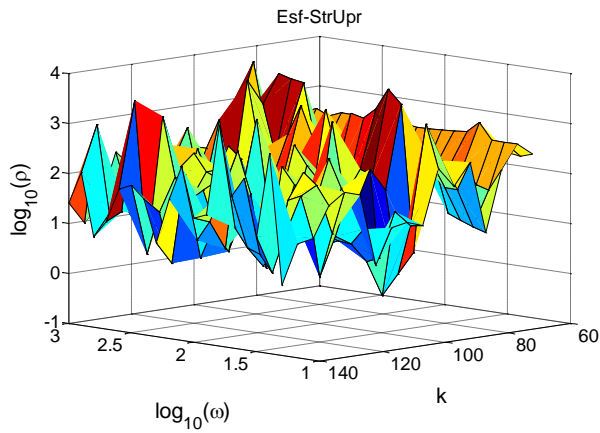


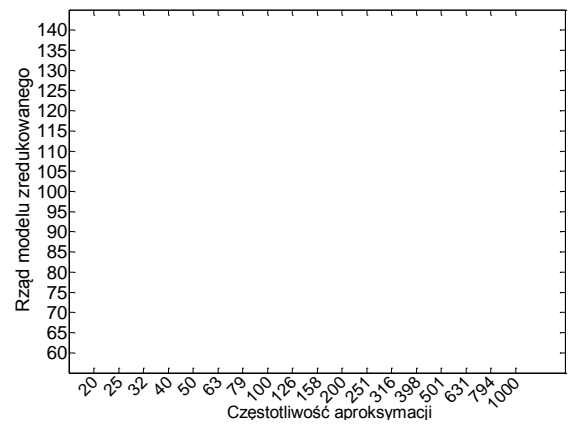
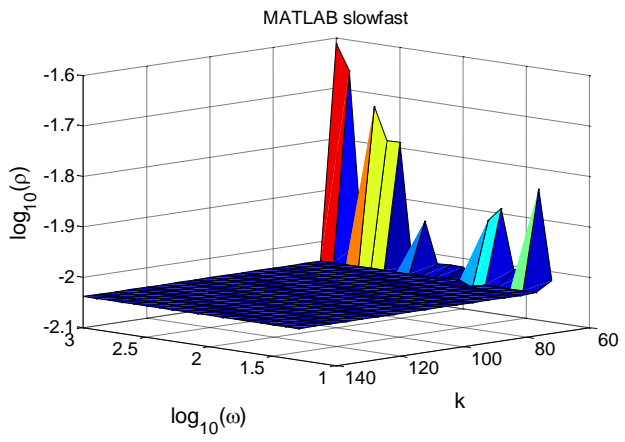
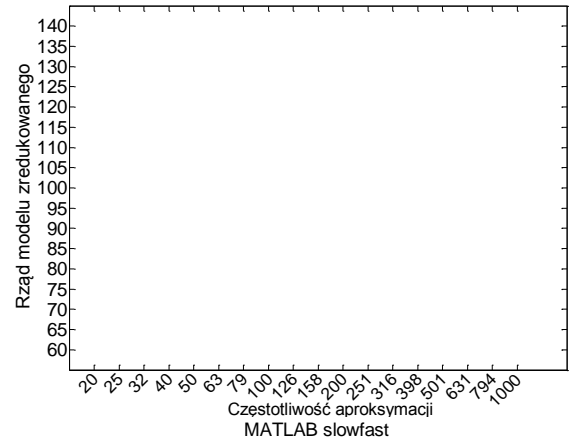
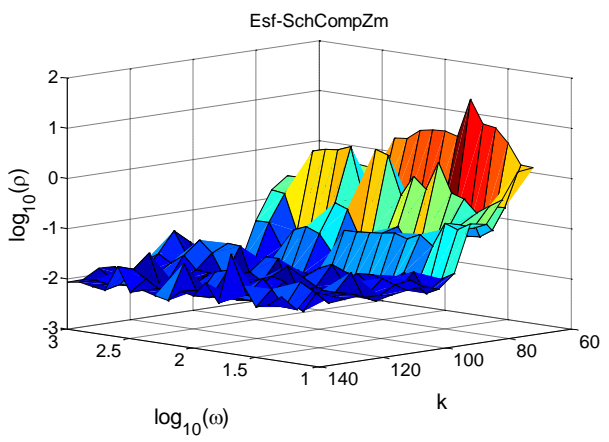
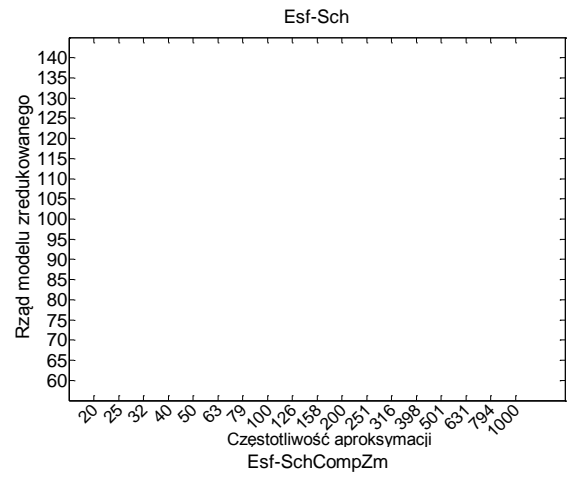
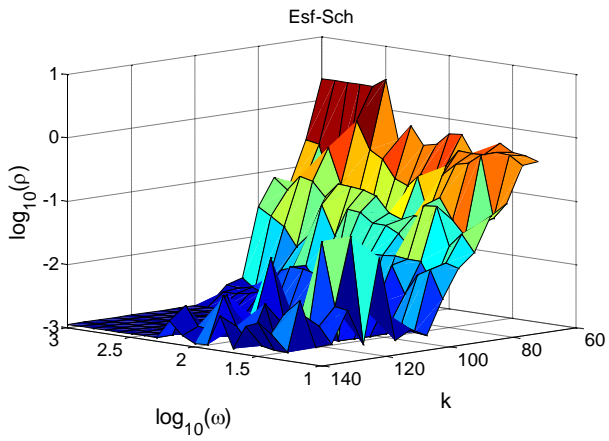
StrUpr



SchCompZm

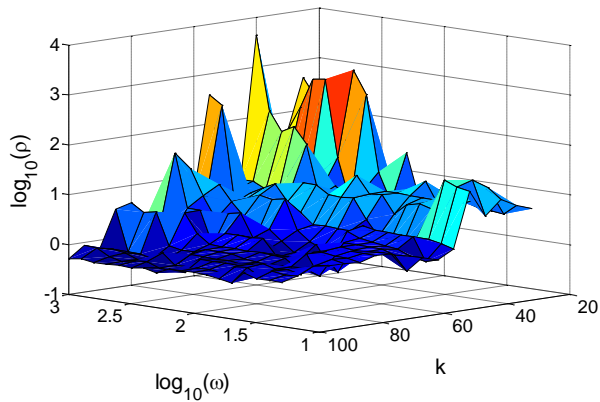




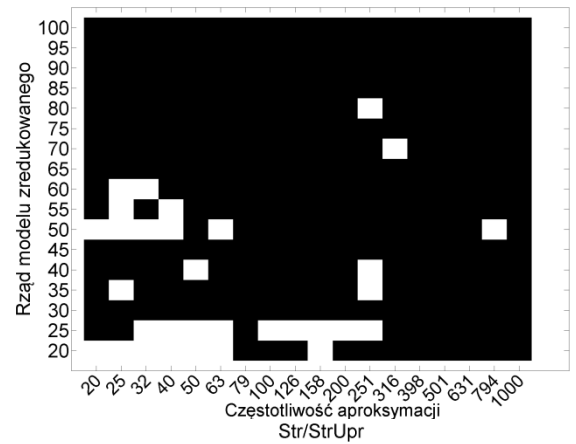


Dodatek H. – wykresy błędów i mapy stabilności dla redukcji powiązanej z dekompozycją dwupoziomową

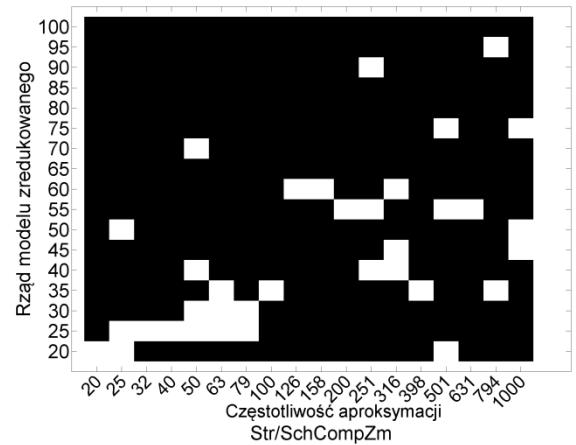
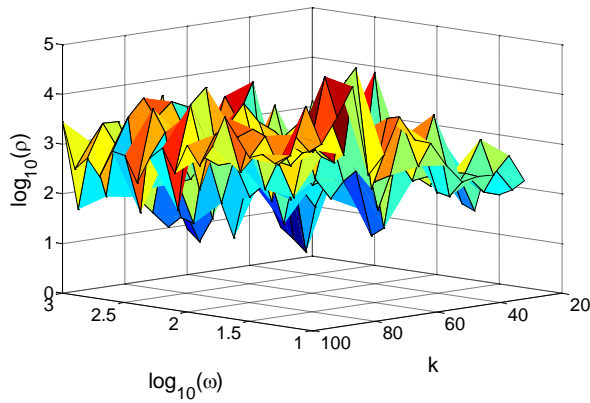
Błąd redukcji:
Str/SchComp



Mapa stabilności:
Str/SchComp



Str/StrUpr



Str/SchCompZm

