POLSKA AKADEMIA NAUK
KOMITET ELEKTRONIKI I TELEKOMUNIKACJI

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI

ELECTRONICS AND TELECOMMUNICATIONS QUARTERLY

TOM 48 — ZESZYT 1

WYDAWNICTWO NAUKOWE PWN
WARSZAWA 2002

RADA REDAKCYJNA

Przewodniczący
Prof. dr hab. inż. STEFAN HAHN
czł. koresp. PAN

Członkowie
prof. dr hab. inż. DANIEL JÓZEF BEM — czł. koresp. PAN, prof. dr hab. inż. MICHAŁ BIAŁKO — czł. rzecz. PAN, prof. dr hab. inż. JAN CHOJCAN, prof. dr hab. inż. MAREK DOMAŃSKI, prof. dr hab. inż. ANDRZEJ HAŁAS, prof. dr inż. WŁADYSŁAW MAJEWSKI, prof. dr hab. inż. JÓZEF MODELSKI, prof. dr inż. JERZY OSIOWSKI, prof. dr hab. inż. EDWARD SĘDEK, prof. dr hab. inż. MICHAŁ TADEUSIEWICZ, prof. dr hab. inż. WIESŁAW WOLIŃSKI — czł. koresp. PAN, prof. dr inż. MARIAN ZIENTALSKI

REDAKCJA
Redaktor Naczelny
prof. dr hab. inż. WIESŁAW WOLIŃSKI

Zastępca Redaktora Naczelnego
doc. dr inż. KRYSTYN PLEWKO

Sekretarz Odpowiedzialny
mgr ELŻBIETA SZCZEPANIAK

ADRES REDAKCJI
00-665 Warszawa, ul. Nowowiejska 15/19 Politechnika, pok. 470
Instytut Telekomunikacji, Gmach im. prof. JANUSZA GROSZKOWSKIEGO

Dyżury Redakcji: środy i piątki, godz. 14–16
tel/fax (022) 660 77 37

Telefony domowe: Redaktora Naczelnego: 812 17 65
Zast. Red. Naczelnego: 826 83 41
Sekretarza Odpowiedzialnego: 633 92 52

WYDAWNICTWO NAUKOWE PWN SA
Warszawa, ul. Miodowa 10

Ark. wyd. 13,25 Ark. druk. 10,5	Podpisano do druku w lutym 2002 r.
Papier offset. kl. III 80 g. B-1	Druk ukończono w marcu 2002 r.

SKŁAD I ŁAMANIE: PHOTOTEXT, Warszawa, ul. Chmielna 120
DRUK I OPRAWA: Zakład Poligraficzny Oja Druk, ul. Sporna 2F, Warszawa

Szanowni Autorzy

„Kwartalnik Elektroniki i Telekomunikacji" — Electronics and Telecommunications Quartery jest kontynuatorem tradycji powstałego 48 lat temu kwartalnika p.t. „Rozprawy Elektrotechniczne".

Kwartalnik jest czasopismem Komitetu Elektroniki i Telekomunikacji Polskiej Akademii Nauk. Wydawany jest przez Wydawnictwo Naukowe PWN SA w Warszawie. Kwartalnik jest czasopismem naukowym, na którego łamach są publikowane artykuły i komunikaty prezentujące wyniki oryginalnych prac teoretycznych i doświadczalnych, a także przeglądowych. Związane są one z szeroko rozumianymi dziedzinami współczesnej elektroniki, telekomunikacji, mikroelektroniki, optoelektroniki, radiotechniki i elektroniki medycznej.

Autorami publikacji są wybitni naukowcy, znani specjaliści o wieloletnim doświadczeniu, a także młodzi badacze — głównie doktoranci.

Artykuły charakteryzują się oryginalnym ujęciem zagadnienia, interesującymi wynikami badań, krytyczną oceną teorii lub metod, omówieniem aktualnego stanu, lub postępu danej gałęzi techniki oraz omówieniem perspektyw rozwojowych. Sposób pisania matematycznej części artykułów zgodny jest z wytycznymi IEC (International Electronics Commision) oraz ISO (International Organization of Standarization).

Wszystkie publikowane w Kwartalniku artykuły są recenzowane przez znanych krajowych specjalitów, co zapewnia że publikacje te są uznawane jako autorski dorobek naukowy. Opublikowanie w kwartalniku wyników prac naukowych zrealizowanych w ramach „GRANTów" Komitetu Badań Naukowych spełnia więc jeden z wymogów stawianych tym pracom.

Czasopismo dociera do wszystkich zajmujących się elektroniką i telekomunikacją krajowych ośrodków naukowych oraz technicznych, a także szeregu instytucji zagranicznych. Jest ponadto prenumerowane przez liczne grono specjalistów i biblioteki.

Każdy Autor otrzymuje bezpłatnie 20 egzemplarzy nadbitek swojego artykułu, co ułatwia przesłanie go do indywidualnych wybranych przez Autora osób i instytucji w kraju lub za granicą. Ułatwia to dodatkowo fakt, że w Kwartalniku są publikowane artykuły w języku angielskim.

Nadesłane do redakcji artykuły są publikowane w terminie około pół roku, w przypadku sprawnej współpracy Autora z Redakcją. Wytyczne dla Autorów dotyczące formy publikacji są zamieszczone w zeszytach Kwartalnika, można je także otrzymać w siedzibie Redakcji.

Artykuły można dostarczać osobiście, lub pocztą pod adresem zamieszczanym na stronie redakcyjnej w każdym zeszycie.

Redakcja

SPIS TREŚCI

W. Pogribny, A. Rajs: Kodowanie i odtwarzanie sygnałlów losowych z wykorzystaniem ekonomicznych modulacji różnicowych 7
M. Gajer: Szeregowanie wieloprocesowrowych zadań okresowych dla trzech dedykowanych procesów 27
D. Kania: Synteza logiczna dla struktur typu PAL wykorzystująca bufory wyjściowe 53
E. Jaszczyszyn, J. Modelski: Synteza paskowych szyków antenowych na podłożu ferroelektrycznym . 67
F. Wysocka-Schillak: Projektowanie filtrów SOI o liniowej fazie z zastosowaniem aproksymacji równomiernie falistej w pasmie przepustowym i średniokwadratowej w pasmie zaporowym . . . 81
B. Stec, A. Dobrowolski, W. Susek: Estymacja wewnętrznego rozkładu temperatury w tkankach biologicznych za pomocą wieloczęstotliwościowego termografu mikrofalowego 101
M. Gajer: Wprowadzenie do zagadnień translacji automatycznej 117
Informacje dla autorów (w jęz. pol.) 161

CONTENTS

W. Pogribny, A. Rajs: Encoding and reproduction of random signals with economical differencial modulations 7
M. Gajer: Multiprocessor periodic tasks scheduling for three dedicated processors 27
D. Kania: Logic synthesis on PAL-based devices containing output buffers 53
E. Jaszczyszyn, J. Modelski: Synthesis of microstrip antenna arrays on ferroelectic substrate 67
F. Wysocka-Schillak: Desing of linear phase fir filters with equiripple passeband and least-squares stopband 81
B. Stec, A. Dobrowolski, W. Susek: Estimation of internal distribution of temperature inside biological tissues by means of multiferequency microwave thermograph 101
M. Gajer: The introduction to machine translation 117
Information for the Authors (w jęz. angl.) 161

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2002, 48, z. 1, ss. 7–25

Kodowanie i odtwarzanie sygnałów losowych z wykorzystaniem ekonomicznych modulacji różnicowych

WŁODZIMIERZ POGRIBNY, ARKADIUSZ RAJS

Instytut Telekomunikacji, Akademia Techniczno-Rolnicza
Al. Prof. Sylwestra Kaliskiego 7,85-796 Bydgoszcz
e-mail: pohry@atr.bydgoszcz.pl
arajs@atr.bydgoszcz.pl

Otrzymano 2001.07.02
Autoryzowano 2001.10.01

Praca jest poświęcona zbadaniu algorytmów różnicowego kodowania z równomiernym próbkowaniem sygnałów losowych. Uwagę skupiono na algorytmach koderów z krokami kwantowania proporcjonalnymi do potęgi liczby 2. Algorytmy te umożliwiają uproszczenie operacji mnożenia, sprowadzając je do zwykłego przesunięcie kodu, co jest szczególnie ważne przy cyfrowej obróbce sygnałów. Celem pracy jest opracowanie i zbadanie dokładności ekonomicznych i efektywnych algorytmów kodowania sygnałów losowych z wyżej wymienionymi krokami na podstawie różnicowej PCM (DPCM) oraz jednobitowej adaptacyjnej modulacji delta (ADM) ze zmienną stromością w dziedzinach czasowej i częstotliwościowej oraz propozycja struktur koderów i dekoderów z wykorzystaniem opracowanych algorytmów. Jako sygnały wejściowe wybrano szum różowy oraz sygnał mowy (słowa w języku polskim według podziału na fonemy). Wybór algorytmów opierał się na minimalizacji błędów średniokwadratowych w dziedzinach czasowej i częstotliwościowej, oraz ocenie subiektywnej.

Słowa kluczowe: modulacja różnicowa, sygnał mowy, cyfrowa obróbka sygnałów.

1. WPROWADZENIE

Sygnały opisywane procesami losowymi niestacjonarnymi, w tym sygnał mowy, są trudne do przetwarzania [1,2]. Bezpośrednie wykorzystanie modulacji PCM do kodowania mowy prowadzi do dużych przepływności strumienia danych w systemach telekomunikacyjnych. W związku z tym w obecnej chwili na świecie trwają prace nad realizacją systemów, które wymagają z jednej strony bardzo krótkich kodów, a z drugiej zabezpieczenia zadanej dokładności. Takimi kodami bez wątpienia dysponują modulacje

różnicowe, do których należy modulacja delta (DM). Wszystkie rodzaje DM zapewniają ekonomiczną cyfryzację sygnałów i dlatego znajdują zastosowanie w telekomunikacji i informatyce [3], co powoduje potrzebę badań odpowiednich algorytmów. Praktycznymi korzyściami z kodowania sygnału za pomocą metod różnicowych jest zysk kompresji strumienia danych oraz odporność na zakłócenia. Wykorzystanie różnicowych formatów jest więc uzasadnione, jednak do dzisiaj nie zostały dostatecznie zoptymalizowane algorytmy przetwarzania sygnałów mowy na podstawie DPCM z krokami kwantowania ze zbioru $\{\lfloor 2^k \rfloor\}$oraz jednobitowej adaptacyjnej DM (ADM) ze zmienną stromością. Istniejące metody różnicowego kodowania sygnałów z dyskretyzacją (próbkowaniem) nierównomierną pozwalają na uzyskanie zadowalającej dokładności przetwarzania, jednak nie mogą być wykorzystywane do obróbki sygnałów [4]. W tej pracy skupimy się głównie na analizie i optymalizacji algorytmów funkcjonowania koderów ADM i DPCM z dyskretyzacją równomierną i krokami kwantowania proporcjonalnymi do potęgi liczby 2 przeznaczonych do kodowania sygnałów typu szumu różowego i mowy.

Celem pracy jest opracowanie i zbadanie w dziedzinach czasowej i częstotliwościowej dokładności ekonomicznych i efektywnych algorytmów kodowania sygnałów losowych (szumu różowego i sygnału mowy) na podstawie modernizowanej DPCM oraz jednobitowej ADM ze zmienną stromością, a także przedstawienie struktur koderów i dekoderów z wykorzystaniem opracowanych algorytmów.

2. WYBÓR EKONOMICZNYCH ALGORYTMÓW MODULACJI RÓŻNICOWYCH

2.1 PARAMETRY MODULACJI DELTA

Zasada działania popularnych rodzajów modulacji DM takich jak; liniowa DM, DPCM, ADPCM, ADM i wielu innych ze sprzężeniem zwrotnym jest następująca. Sygnał aproksymacji $\hat{x}(t)$ jest generowany w sprzężeniu zwrotnym kodera DM. Różnica α_i pomiędzy kolejnymi próbkami sygnału wejściowego $\{x_m\}$ i aproksymacji $\{\hat{x}_m\}$w i-tej chwili wyraża się zależnością

$$\alpha_i = x_i - \hat{x}_i \tag{1}$$

Kiedy ustalamy powiązanie różnic z kodami DM (krokami kwantowania) musimy wziąć pod uwagę, że DM oprócz szumów kwantowania wprowadza także specyficzny typ błędów powiązanych z przeciążeniem kodera wywołanego stromością sygnału. Przy wyznaczaniu parametrów DM takich jak częstotliwość dyskretyzacji i wartości kroków kwantowania należy minimalizować oba rodzaje błędów. Warunek pojawienia się drugiego z tych błędów powiązanego z przeciążeniem kodera jest następujący:

$$\alpha_{\max} > \varepsilon s_{\max} \pm \varepsilon, \tag{2}$$

$$\alpha_{\max} = \max_{i=\overline{1,N}} |x_i - \hat{x}_i|, \qquad \mathrm{N} = \mathrm{ENT}(\Theta/\mathrm{T}), \tag{3}$$

gdzie: $\alpha_{\max}$ – maksymalna różnica między próbkami sygnałów wejściowego i aproksymacji, $s_{\max}$, $s_{\min}$ – maksymalny i minimalny bezwymiarowy niezerowy krok kwantyzacji DM, ε – wielkość bezwzględna minimalnego kroku, N – liczba próbek, Θ – czas realizacji sygnału, T – okres dyskretyzacji. Kody DM zapisane w formie kroków kwantowania $\{s_i\}$ mogą być zgodne z różnicami przy założeniu braku przeciążenia kodera wywołanego stromością. Wtedy aproksymacja sygnału DM$\{\hat{x}_i\}$ jest związana z sygnałem wejściowym $\{x_i\}$ według następującego wyrażenia:

$$\hat{x}_i = x_{i-1} \pm \psi_i, \qquad \alpha_i = x_i - x_{i-1} \pm \psi_i \approx \nabla x_i \tag{4}$$

gdzie ψ – błąd kwantyzacji. Jeżeli nie występuje przeciążenie kodera DM wywołane stromością sygnału na wejściu wtedy ma miejsce: $\forall|\psi| \in [0, \varepsilon]$. Określmy za pomocą (4) zależność między przyrostem sygnału a wartością kroku kwantyzacji $s_r^{(x)}$:

$$\nabla x_i = \varepsilon s_i^{(x)} \pm \psi_i \tag{5}$$

Sygnał aproksymacji jest formowany poprzez kroki kwantowania $\{\varepsilon s_k^{(x)}\}$ w sposób

$$\hat{x}_r = \varepsilon \sum_{k=0}^{r} s_k^{(x)} \approx x_0 + \varepsilon \sum_{k=1}^{r} s_k^{(x)} \tag{6}$$

Dla rozpatrywanych w tej pracy algorytmów koderów ADM i DPCM ma miejsce $\alpha_i \in [0, \varepsilon s_{\max} + \varepsilon]$.

Podstawą we wszystkich modulacjach różnicowych z dyskretyzacją równomierną jest wybór częstotliwości dyskretyzacji i kroków kwantowania. Maksymalny przyrost sygnału ∇x_{max} jest zależny od częstotliwości próbkowania. Ogólne wyrażenie służące do wyznaczenia częstotliwości dyskretyzacji na podstawie różnic sygnału otrzymanych a'posteriori ma postać:

$$T^{-1} \geq [x'(t)]_{\max} / \nabla x_{\max}, \tag{7}$$

gdzie $[x'(t)]_{\max}$ – maksymalna pochodna sygnału. Wyrażenie (7) nie może być użyte do przedstawiania sygnału w trybie czasu rzeczywistego, ponieważ zazwyczaj nie ma bezpośredniej informacji o pochodnej sygnału. Określenie maksymalnej pochodnej sygnału można zrealizować pośrednio na podstawie widma poprzednich realizacji sygnału. Skorzystamy z szeregu Fouriera rzeczywistego sygnału:

$$x(t) \approx \bar{x} + 2 \sum_{k=1}^{N/2} |X_k| \cos(2\pi f_k t + \varphi_k) \tag{8}$$

gdzie: $\bar{x}$ – średnia wartość sygnału, $\{|X_k|\}$ – moduł składnika Fouriera znalezionego na podstawie DFT sygnału, $f_k = k/\Theta$ – częstotliwość k – tej harmonicznej, φ_k – faza k-tej harmonicznej. Wtedy dla pochodnej sygnału:

$$x'(t) \approx 4\pi \sum_{k=1}^{N/2} |X_k| f_k [-\sin(2\pi f_k t + \varphi_k)] \tag{9}$$

W przypadku, kiedy wszystkie sinusy są równe -1, wyrażenie to daje nam wartość idealnej i praktycznie niespotykanej maksymalnej pochodnej:

$$[x'(t)]^{(i)}_{\max(i)} \approx 4\pi \sum_{k=1}^{N/2} |X_k| f_k. \tag{10}$$

Bezpośrednie podstawienie (10) do (7) prowadzi do wyboru za wysokiej częstotliwości dyskretyzacji. Symulacje komputerowe pokazały, że w większości przypadków maksymalna pochodna jest formowana z 2, 3 składników, które dają w rezultacie maksymalny iloczyn modułów składników z ich częstotliwościami. Przy wykorzystaniu takiego podejścia możemy wyrazić częstotliwość dyskretyzacji dla różnicowych formatów za pomocą (11) pod warunkiem, że $\nabla x_{\max} = \varepsilon s_{\max}$:

$$f_D = T^{-1} = 4\pi \left(\overline{|X_i|} \cdot f_i + \overline{|X_r|} f_r\right)_{\max} / \varepsilon s_{\max} \tag{11}$$

gdzie $\left(\overline{|X_i|} \cdot f_i + \overline{|X_r|} f_r\right)_{\max}$ jest maksymalną sumą dwóch iloczynów uśrednionych współczynników Fouriera z ich częstotliwościami.

Korzystając z tego wyrażenia i uwzględniając $2|X_k| = U_k$, można zapisać na podstawie [7] algorytm wyboru w czasie rzeczywistym częstotliwości dyskretyzacji ADM i 3-bitowej DPCM, z maksymalnym krokiem 4ε następująco:

$$f_D = \frac{2\pi \left(U_i^{(x)} f_i + U_r^{(x)} f_r\right)_{\max}}{4\pi}, \tag{12}$$

gdzie $U_k^{(x)}$ – amplituda k- tej składowej sygnału $x(t)$ otrzymana a'priori.

Wybór kroków kwantowania jest opisany zależnościami (13) i (14). Zbyt mała wartość ε jest źródłem błędu przeciążenia, natomiast zbyt duża wartość ε prowadzi do zwiększenia szumów kwantowania.

Maksymalny krok kwantowania możemy wyrazić następująco:

$$\varepsilon s_{\max} = 4\varepsilon \geq \max_{i=\overline{1,N}} |x_i - x_{i-1}| \tag{13}$$

Minimalny krok kwantowania [5] określa wyrażenie:

$$\varepsilon s_{\min} = \varepsilon \leq \pi \frac{U_g^{(x)}}{\mu} \tag{14}$$

gdzie $U_g^{(x)}$ – wartość górnej harmonicznej widma, μ – iloraz częstotliwości próbkowania DM i PCM. Przy określonej częstotliwości dyskretyzacji można z wyrażeń (11) i (12) wyznaczyć maksymalny przyrost oraz maksymalny krok.

2.2. WYBÓR ALGORYTMU ADM

Przeprowadzone badania pokazały, że w celu uzyskania najlepszych rezultatów kodowania, należy zastosować inny typ algorytmu kodera dla sygnału będącego reprezentacją szumu różowego z równomiernym widmem amplitudowym, a inny dla sygnału mowy.

2.2.1. Wybór i dokładność algorytmu kodera dla sygnału typu szum różowy

Przy przeprowadzaniu badań kodera DM przetwarzającego sygnał o równomiernym widmie wykorzystujemy tę samą realizację szumu różowego, którego wszystkie składowe częstotliwościowe są równe, ich częstotliwości należą do zbioru liczb naturalnych, a przesunięcia fazowe są losowe. W poniższej tabeli zestawiono kilkanaście przebadanych algorytmów pracy koderów ADM i DPCM i ich błędy średniokwadratowe przy jednakowej częstotliwości dyskretyzacji i kroku ε. Kroki kwantowania dobierano następująco:

- dla koderów ADM kroki kwantowania wybierano ze zbioru $\{-8\varepsilon, -4\varepsilon, -2\varepsilon, -\varepsilon, \varepsilon, 2\varepsilon, 4\varepsilon, 8\varepsilon\}$,
- dla koderów DPCM 3 bitowych ze zbiorów:
  klasycznego: $\{-3\varepsilon, -2\varepsilon, -\varepsilon, 0, \varepsilon, 2\varepsilon, 3\varepsilon\}$,
  modyfikowanego: $\{-4\varepsilon, -2\varepsilon, -\varepsilon, 0, \varepsilon, 2\varepsilon, 4\varepsilon\}$,
- dla koderów DPCM 4 bitowych ze zbiorów kroków kwantowania:
  klasycznego: $\{-7\varepsilon, -6\varepsilon, -5\varepsilon, -4\varepsilon, -3\varepsilon, -2\varepsilon, -\varepsilon, 0, \varepsilon, 2\varepsilon, 3\varepsilon, 4\varepsilon, 5\varepsilon, 6\varepsilon, 7\varepsilon\}$,
  modyfikowanego: $\{-64\varepsilon, -32\varepsilon, -16\varepsilon, -8\varepsilon, -4\varepsilon, -2\varepsilon, -\varepsilon, 0, \varepsilon, 2\varepsilon, 4\varepsilon, 8\varepsilon, 16\varepsilon, 32\varepsilon, 64\varepsilon\}$
- dla kodera LDM ze zbioru kroków $\{-\varepsilon, \varepsilon\}$.

Optymalizacja algorytmów polega na minimalizacji błędu średniokwadratowego aproksymacji w dziedzinie czasu, który jest opisany następującym wyrażeniem:

$$\sigma_A = \sqrt{\frac{1}{N} \sum_{r=0}^{N} (x_r - \hat{x}_r)^2} \tag{15}$$

Błąd ten zależy od wyboru częstotliwości dyskretyzacji, wartości ε oraz algorytmu pracy kodera który określa kolejność kroków kwantowania przy narastaniu i obniżeniu sygnału.

W tabeli 1 zestawiono wyniki symulacji komputerowych obrazujące wartość błędu średniokwadratowego dla poszczególnych algorytmów koderów.

Tabela 1

Dokładność algorytmów koderów dla szumu różowego

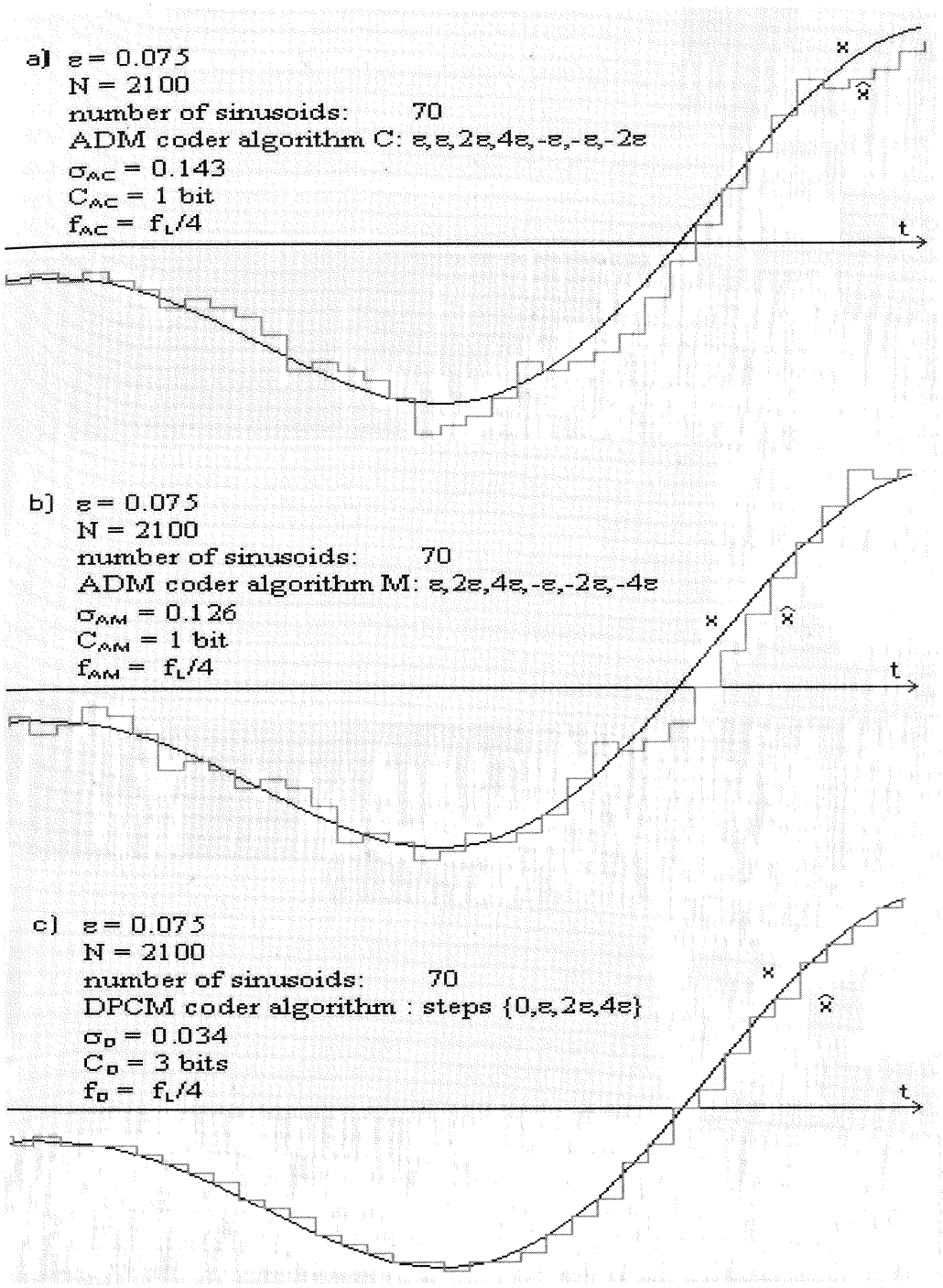
Dokładność algorytmów koderów dla szumu różowego przy $f_A = 4f_L = f_D$, $\varepsilon_A = \varepsilon_L = \varepsilon_D = 0{,}075$ Ug,					
N°	**ADM (1 bit)**		**LDM (1 bit)**		
	Algorytm	σ_A	Algorytm*		σ_L
1.	$\varepsilon, 2\varepsilon, 4\varepsilon, -\varepsilon, -2\varepsilon, -4\varepsilon$	0.126	$\{\varepsilon, -1\varepsilon\}$		0.661
2.	$\varepsilon, \varepsilon, 2\varepsilon, 2\varepsilon, 4\varepsilon, -\varepsilon, -\varepsilon, -2\varepsilon, -2\varepsilon, -4\varepsilon$	0.129			
3.	$\varepsilon, 2\varepsilon, 4\varepsilon, -2\varepsilon, -\varepsilon, -2\varepsilon, -4\varepsilon$	0.129	**DPCM (3 bity)**		
4.	$\varepsilon, 2\varepsilon, 4\varepsilon, -2\varepsilon, -4\varepsilon$	0.140	Algorytm		σ_D
5.	$\varepsilon, \varepsilon, 4\varepsilon, 4\varepsilon, -\varepsilon, -\varepsilon, 4\varepsilon$	0.140	1.	$\{0, 1\varepsilon, 2\varepsilon, 4\varepsilon\}$	0.034
6.	$\varepsilon, \varepsilon, 2\varepsilon, 4\varepsilon, -\varepsilon, -\varepsilon, -2\varepsilon, -4\varepsilon$	0.143	2.	$\{0, 1\varepsilon, 2\varepsilon, 3\varepsilon\}$	0.104
7.	$\varepsilon, \varepsilon, 2\varepsilon, 4\varepsilon, -2\varepsilon, -\varepsilon, -\varepsilon, -2\varepsilon, -4\varepsilon$	0.144			
8.	$\varepsilon, \varepsilon, 2\varepsilon, 2\varepsilon, 4\varepsilon, 4\varepsilon, -2\varepsilon, -2\varepsilon, -4\varepsilon$	0.145	**DPCM (4 bity)**		
9.	$\varepsilon, 4\varepsilon, 4\varepsilon, -2\varepsilon, -4\varepsilon, 2\varepsilon, -\varepsilon, \varepsilon, 4\varepsilon$	0.153	Algorytm		σ_D
10.	$\varepsilon, 4\varepsilon, -\varepsilon, -4\varepsilon$	0.156	1.	$\{0, 1\varepsilon, 2\varepsilon, 3\varepsilon, 4\varepsilon, ..., 7\varepsilon\}$	0.014 dla $\varepsilon = 0.048$
11.	$\varepsilon, \varepsilon, 2\varepsilon, 4\varepsilon, -2\varepsilon, -4\varepsilon$	0.161	2.	$\{0, 1\varepsilon, 2\varepsilon, 4\varepsilon, ..., 64\varepsilon\}$	0.025 dla $\varepsilon = 0.017$
12.	$\varepsilon, \varepsilon, 2\varepsilon, 2\varepsilon, 4\varepsilon, 4\varepsilon, 8\varepsilon, -4\varepsilon, -4\varepsilon, -8\varepsilon$	0.164			
13.	$\varepsilon, \varepsilon, 4\varepsilon, -2\varepsilon, -4\varepsilon, 2\varepsilon, -\varepsilon, \varepsilon, 4\varepsilon$	0.164			
14.	$\varepsilon, 2\varepsilon, 4\varepsilon, 8\varepsilon, -\varepsilon, -2\varepsilon, -4\varepsilon, -8\varepsilon$	0.169			
15.	$\varepsilon, \varepsilon, 2\varepsilon, 2\varepsilon, 4\varepsilon, 4\varepsilon, 8\varepsilon, -\varepsilon, -\varepsilon, -2\varepsilon$	0.169			
16.	$\varepsilon, \varepsilon, 2\varepsilon, 4\varepsilon, 8\varepsilon, -\varepsilon, -\varepsilon, -2\varepsilon, -4\varepsilon, -8\varepsilon$	0.175			
17.	$\varepsilon, 2\varepsilon, 4\varepsilon, 8\varepsilon, -4\varepsilon, -2\varepsilon, -\varepsilon, \varepsilon, 2\varepsilon, 4\varepsilon, 8\varepsilon$	0.179			
18.	$\varepsilon, \varepsilon, 2\varepsilon, 4\varepsilon, 8\varepsilon, -4\varepsilon, -8\varepsilon, 4\varepsilon, 8\varepsilon$	0.195			
19.	$\varepsilon, 2\varepsilon, 4\varepsilon, 8\varepsilon, -4\varepsilon, -8\varepsilon$	0.198			
20.	$\varepsilon, \varepsilon, 2\varepsilon, 4\varepsilon, 8\varepsilon, -4\varepsilon, -2\varepsilon, -\varepsilon$	0.202			
21.	$\varepsilon, \varepsilon, 2\varepsilon, 4\varepsilon, \varepsilon, 2\varepsilon, 4\varepsilon, -2\varepsilon, -4\varepsilon$	0.286			

Na rys.1 przedstawione zostały wykresy czasowe obrazujące pracę trzech wybranych koderów. Na wykresach tych znajduje się przebieg sygnału wejściowego $x(t)$ oraz sygnału aproksymacji $\hat{x}(t)$. Na podstawie symulacji komputerowych ustalono, że dla kodowania sygnału typu szumu różowego najlepszą pracę koderów ADM uzyskuje się wykorzystując kroki kwantowania ze zbioru $\{-4\varepsilon, -2\varepsilon, -\varepsilon, \varepsilon, 2\varepsilon, 4\varepsilon\}$ (tabela1).

Tabela 1

5 Ug,
8
7

rzech
wego
talo-
ADM
$4\varepsilon\}$



Rys. 1. Fragment realizacji sygnału typu szum różowy zakodowanego przy użyciu klasycznej ADM (a), modyfikowanej ADM (b), i DPCM (c)

2.2.2. *Wybór i dokładność algorytmu kodera dla sygnału mowy*

W celu zbadania dokładności ekonomicznych i efektywnych algorytmów kodowania mowy w dziedzinie czasowej i częstotliwościowej wybraliśmy słowa wypowiadane w języku polskim głosem męskim, żeńskim i dziecięcym. Dobór reprezentatywnych słów został przeprowadzony wg kryterium podziału na fonemy [8]. Wybierano słowa zawierające fonemy należące do następujących grup:

- samogłoski (np. mur, rok, lis),
- spółgłoski zwarte (np. dom, pas, kino),
- spółgłoski trące (np. frak, kosa, siano),
- spółgłoski zwarto-trące (np. praca, sadza, działo),
- spółgłoski nosowe (np. matka, bank, ręka),
- spółgłoski boczne (np. lody),
- samogłoski niesylabiczne (np. jodła, ławka),
- spółgłoska drżąca (np. ryba).

Badaniu poddano kilkanaście różnych algorytmów koderów ADM oraz cztery algorytmy koderów DPCM (3 i 4 bitowy). Wyboru najlepszego algorytmu dokonaliśmy na podstawie wartości stosunku sygnał szum (SNR) w dziedzinie czasowej oraz błędu średniokwadratowego w dziedzinie częstotliwościowej. Wartość SNR jest obliczana na podstawie następującego wyrażenia:

$$SNR = 10 \log_{10} \frac{P_x}{P_A} \tag{16}$$

gdzie: $P_x = \frac{1}{N} \sum_{r=0}^{N-1} x_r^2$, $P_A = \frac{1}{N} \sum_{r=0}^{N-1} (x_r - \hat{x}_r)^2$

Według naszych badań najlepsze rezultaty przy kodowaniu sygnału mowy (korzystając z kodera ADM) uzyskiwaliśmy przy wykorzystaniu algorytmu o postaci: $\{\varepsilon, \varepsilon, 2\varepsilon, 4\varepsilon, 8\varepsilon, -4\varepsilon, -8\varepsilon, 4\varepsilon, 8\varepsilon\}$, co pokazuje tabela 2. Jest to algorytm „symetryczny", a następstwo kroków w tym algorytmie jest następujące: $\varepsilon \to \varepsilon$, $\varepsilon\varepsilon \to 2\varepsilon$, $2\varepsilon \to 4\varepsilon$, $4\varepsilon \to 8\varepsilon$, $8\varepsilon \to 8\varepsilon$, $-\varepsilon \to \varepsilon$, $-2\varepsilon \to \varepsilon$, $-4\varepsilon \to 2\varepsilon$, $-8\varepsilon \to 4\varepsilon$. Algorytm ten wiąże reakcję na wartość sygnału z wielkością ostatniego kroku kwantowania, i tak na przykład po kroku 4ε może nastąpić krok 8ε (jeśli sygnał rośnie) lub -2ε (jeśli sygnał maleje). Maksymalnym krokiem jest krok 8ε, a minimalnym -8ε. Fragment kodu źródłowego programu komputerowego szczegółowo przedstawiający algorytm tego kodera zawiera Dodatek A. Lepsze rezultaty otrzymywaliśmy przy wykorzystaniu koderów DPCM (3 bitowego modyfikowanego i 4 bitowych), jednak należy zaznaczyć, że następuje tu zwiększenie wykorzystywanego pasma. Tabela 3 pokazuje wartości SNR algorytmu kodera #18 w dziedzinie czasowej w zależności od zmiany kroku kwantowania ε i częstotliwości próbkowania f_{ADM}.

Tabela 2

Dokładność zbadanych algorytmów koderów dla sygnału mowy

Dokładność algorytmów koderów dla sygnału mowy przy $f_A = f_D = f_L$, algorytm LDM przy częstotliwości próbkowania równej f_A								
No	**ADM (1 bit)**				**LDM (1 bit)**			
	Algorytm	σ_A	ε	SNR [dB]	Algorytm*	σ_L	ε	SNR [dB]
1.	ε, 2ε, 4ε, $-\varepsilon$, -2ε, -4ε	3.26	2.17	17.35	$\{1\varepsilon, -1\varepsilon\}$	4.69	4.85	14.20
2.	ε, ε, 2ε, 2ε, 4ε, $-\varepsilon$, $-\varepsilon$, -2ε, -2ε, -4ε	2.90	2.86	18.36				
3.	ε, 2ε, 4ε, -2ε, $-\varepsilon$, -2ε, -4ε	3.23	2.57	17.42	**PCM (3 bity)**			
4.	ε, 2ε, 4ε, -2ε, -4ε	3.28	2.80	17.30		σ_D		SNR [dB]
5.	ε, ε, 4ε, 4ε, $-\varepsilon$, $-\varepsilon$, 4ε	3.07	2.85	17.87		23.15		3.67
6.	ε, ε, 2ε, 4ε, $-\varepsilon$, $-\varepsilon$, -2ε, -4ε	2.80	2.94	18.66				
7.	ε, ε, 2ε, 4ε, -2ε, $-\varepsilon$, $-\varepsilon$, -2ε, -4ε	2.77	2.95	18.78	**DPCM (3 bity)**			
8.	ε, ε, 2ε, 2ε, 4ε, 4ε, -2ε, -2ε, -4ε	2.83	2.95	18.57	Algorithm	σ_D	ε	SNR [dB]
9.	ε, 4ε, 4ε, -2ε, -4ε, 2ε, $-\varepsilon$, ε, 4ε	4.31	1.64	14.93	1. $\{0,1\varepsilon,2\varepsilon,3\varepsilon\}$	1.55	4.50	23.83
10.	ε, 4ε, $-\varepsilon$, -4ε	4.39	1.53	14.77	2. $\{0,1\varepsilon,2\varepsilon,4\varepsilon\}$	1.26	3.70	25.58
11.	ε, ε, 2ε, 4ε, -2ε, -4ε	2.77	2.95	18.78				
12.	ε, ε, 2ε, 2ε, 4ε, 4ε, 8ε, -4ε, -4ε, -8ε	2.64	2.58	19.20	**DPCM (4 bity)**			
13.	ε, ε, 4ε, -2ε, -4ε, 2ε, $-\varepsilon$, ε, 4ε	3.04	2.70	17.97	Algorithm	σ_D	ε	SNR [dB]
14.	ε, 2ε, 4ε, 8ε, $-\varepsilon$, -2ε, -4ε, -8ε	2.93	1.65	18.27	1. $\{0,1\varepsilon, 2\varepsilon, 3\varepsilon, \ldots 7\varepsilon\}$	0.84	2.50	29.11
15.	ε, ε, 2ε, 2ε, 4ε, 4ε, 8ε, $-\varepsilon$, $-\varepsilon$, -2ε	2.80	2.80	18.66	2. $\{0,1\varepsilon, 2\varepsilon, 4\varepsilon, \ldots, 64\varepsilon\}$	0.55	1.00	32.75
16.	ε, ε, 2ε, 4ε, 8ε, $-\varepsilon$, $-\varepsilon$, -2ε, -4ε, -8ε	2.72	2.26	18.93				
17.	ε, 2ε, 4ε, 8ε, -4ε, -2ε, $-\varepsilon$, ε, 2ε, 4ε, 8ε	2.71	1.50	18.96				
18.	ε, ε, 2ε, 4ε, 8ε, -4ε, -8ε, 4ε, 8ε	2.51	1.41	19.63				
19.	ε, 2ε, 4ε, 8ε, -4ε, -8ε	2.81	1.56	18.63				
20.	ε, ε, 2ε, 4ε, 8ε, -4ε, -2ε, $-\varepsilon$	2.66	2.36	19,11				
21.	ε, ε, 2ε, 4ε, ε, 2ε, 4ε, -2ε, -4ε	2.86	2.94	18.48				

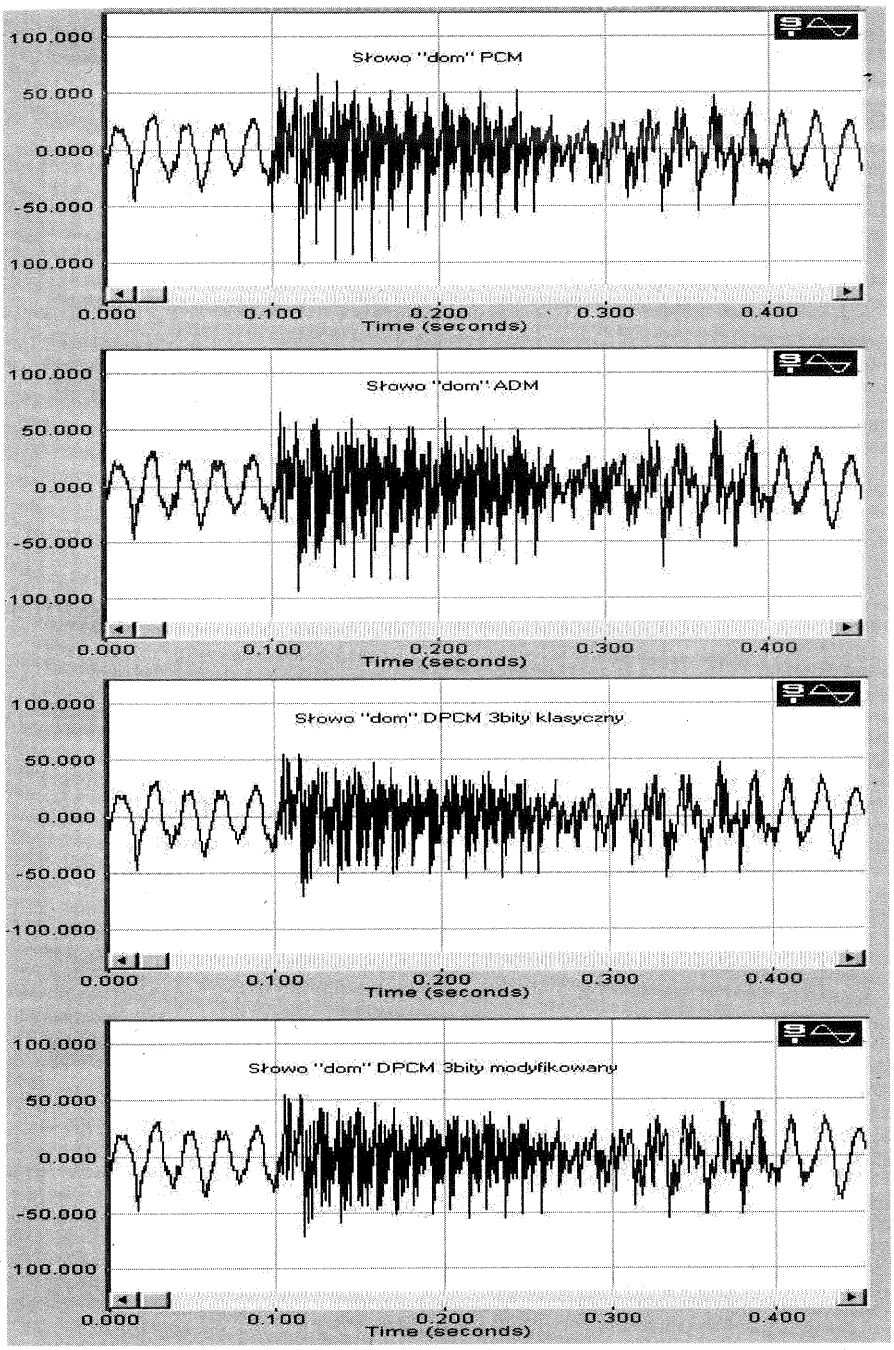
Tabela 3

SNR algorytmu kodera #18 w dziedzinie czasowej w zależności od zmiany kroku kwantowania ε i częstotliwości próbkowania f_{ADM}

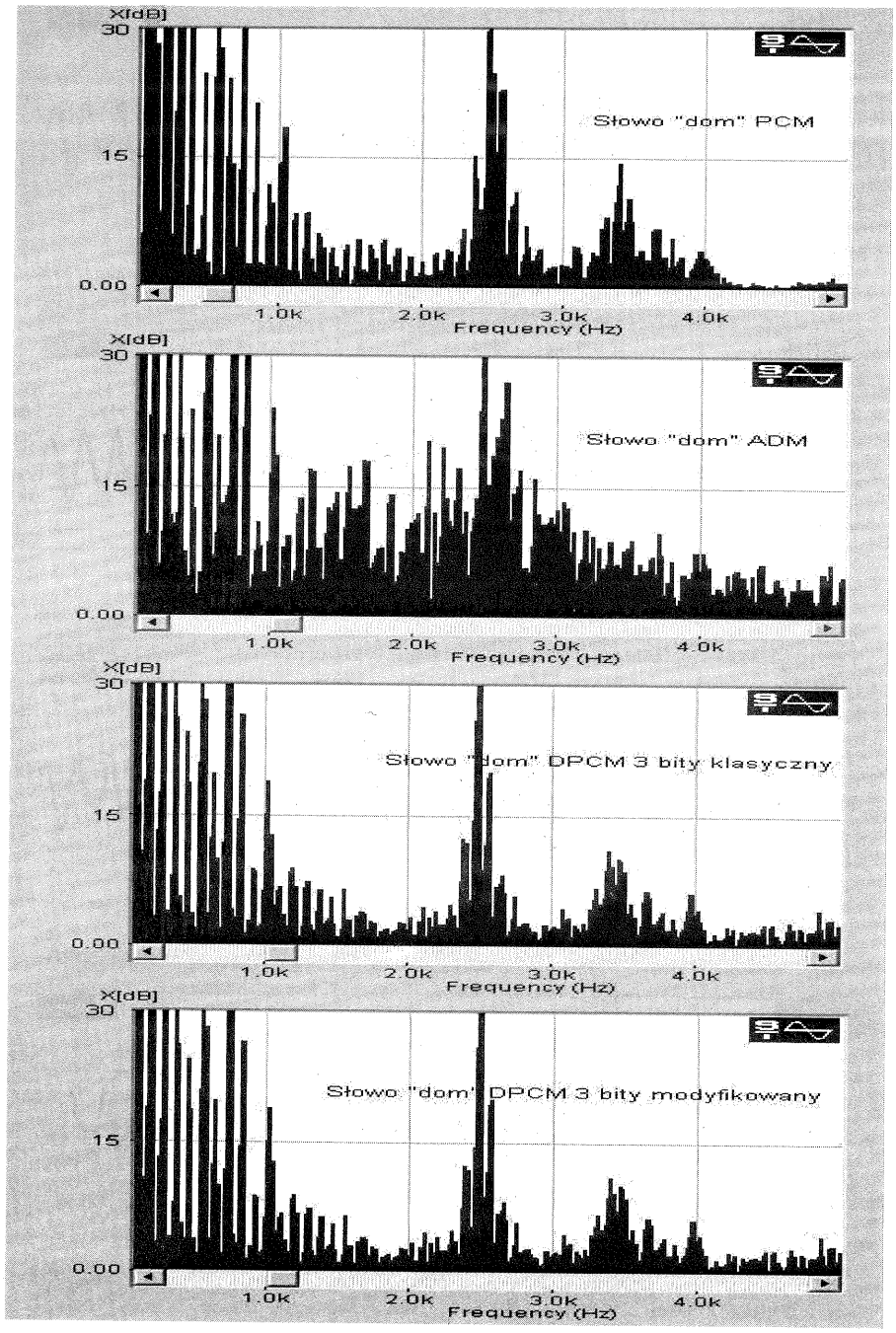
Koder #18 Sygnał mowy, głos męski, słowo „dom''											
f_A [Hz]	ε	σ_A	SNR [dB]	f_A [Hz]	ε	σ_A	SNR [dB]	f_A [Hz]	ε	σ_A	SNR [dB]
11025	0.5	12.21	6.04	22050	0.5	7.98	9.58	44100	0.5	5.16	13.37
	0.8	10.91	7.02		0.8	6.47	11.40		0.8	3.65	16.37
	1.0	10.49	7.37		1.0	5.68	12.54		1.0	3.26	17.35
	1.41	9.37	8.34		1.41	4.78	14.03		1.41	2.51	19.63
	1.5	9.23	8.48		1.5	4.88	13.86		1.5	2.78	18.78
	1.8	8.99	8.70		1.8	4.59	14.39		1.8	2.82	18.60
	2.0	8.71	8.98		2.0	4.43	14.69		2.0	2.60	19.34
	2.2	8.21	9.49		2.2	4.49	14.58		2.2	2.78	18.75
	2.5	8.45	9.24		2.5	4.38	14.79		2.5	3.00	18.08
	2.8	8.10	9.61		2.8	4.46	14.65		2.8	3.06	17.91
	3.0	8.69	9.00		3.0	4.57	14.43		3.0	3.21	17.50
	3.2	8.49	9.20		3.2	4.63	14.31		3.2	3.43	16.92
	3.31	7.82	9.92		3.31	4.82	13.96		3.31	3.31	17.23
	3.5	8.11	9.59		3.5	4.91	13.80		3.5	3.49	16.76
	3.8	8.55	9.14		3.8	5.06	13.54		3.8	3.72	16.21
	4.0	8.33	9.36		4.0	5.11	13.45		4.0	3.65	16.38
	4.2	8.40	9.29		4.2	5.16	13.37		4.2	3.82	15.98
	4.5	8.51	9.18		4.5	5.29	13.16		4.5	3.89	15.82

Wybrany algorytm został zilustrowany wynikami symulacji komputerowych. Na rysunku 2 przedstawione są przebiegi czasowe sygnału odpowiadającemu słowu „dom'', wypowiadanego głosem męskim i kodowanemu odpowiednio przy pomocy koderów PCM, wybranego przez nas ADM, DPCM klasycznego i DPCM modyfikowanego. Rysunek 3 pokazuje widma odpowiednich sygnałów. Jak widać na rysunku 2 przebiegi czasowe poszczególnych koderów są podobne, jedynie oba kodery DPCM zmniejszają amplitudę sygnału. Mimo to kodery DPCM charakteryzują się nieco lepszą wiernością kodowanego sygnału w porównaniu z koderem ADM.

Widma amplitudowe (rysunek 3) sygnałów są bardziej zróżnicowane niż przebiegi czasowe i trudno na ich podstawie wyciągnąć jednoznaczne wnioski. Jednak posługując



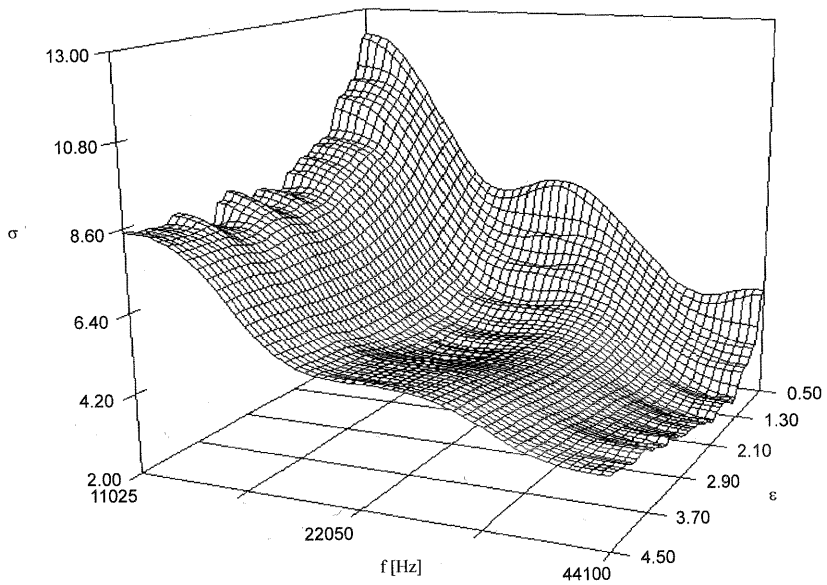
Rys. 2. Przebiegi czasowe przy kodowaniu sygnału mowy (słowo „dom'')



Rys. 3. Widmo sygnału mowy (słowo „dom'')

się miarą subiektywną, można stwierdzić, że przy zastosowaniu na wyjściu dekodera ADM filtru wygładzającego, uzyskamy zadowalającą jakość odtwarzania kodowanego sygnału.

Na rysunku 4 pokazano zależność błędu średniokwadratowego aproksymacji σ_A od częstotliwości próbkowania f_A i wartości kroku kwantowania ε dla wybranego kodera.



Rys. 4. Zależność błędu średniokwadratowego σ_A od częstotliwości próbkowania f_A i wartości kroku kwantowania ε dla kodera #18

Wykres pokazuje, że, jak należało oczekiwać, im większa częstotliwość próbkowania, tym mniejszy błąd średniokwadratowy i w konsekwencji lepsze kodowanie sygnału wejściowego. Przy danej częstotliwości próbkowania zmiana ε, z wyjątkiem wartości bliskiej 0, nieznacznie wpływa na wartość błędu średniokwadratowego. Należy zaznaczyć, że wartość ε wpływająca na najlepszy wynik (najmniejszą wartość błędu średniokwadratowego) zależy od charakteru wypowiadanego sygnału (np. słowa wypowiedziane głośno lub cicho). Z przeprowadzonych badań zobrazowanych na rysunku 4 wynika, że nie istnieje optymalna wartość ε dla wszystkich badanych częstotliwości próbkowania. Na przykład, dla częstotliwości próbkowania f_p 11025 Hz minimalna wartość σ_A ma miejsce przy ε w zakresie 2.5 do 3.5. Analogicznie dla f_p = 22050 Hz wartość ε zawiera się w przedziale 2.0 do 3.0, a dla częstotliwości f_p = 44100 Hz ε powinien zawierać się w granicach między 1.30 a 2.50. Ogólnie można stwierdzić, że przy częstotliwości próbkowania 22050 Hz można uzyskać zadowalające efekty odtwarzania sygnału mowy (porównywalne z tradycyjnym PCM) pod warunkiem zastosowania na wyjściu filtra wygładzającego z odpowiednią charakterystyką częstotliwościową.

3. OPERACJE MATEMATYCZNE NA KODACH MODERNIZOWANYCH ADM I DPCM

Sumowanie kodów ADM i DPCM jest wykonywane typowo jak dla ciągów binarnych. Natomiast przy operacji mnożenia z wykorzystaniem modernizowanych kroków kwantowania ADM i DPCM czyli kroków ze zbioru $\{|2^k|\}$ operacje matematyczne na kodach ADM i DPCM sprowadzają się do zwykłych przesunięć kodu o określoną liczbę pozycji (0, 1, 2, 4...). I tak przedstawiając wartość, która w formacie PCM w postaci kodu binarnego ma postać $h_a = \operatorname{sgn} h_a \cdot |h_a|$ możemy zapisać:

$$\operatorname{sgn} h_a \Rightarrow B_{s.a}^{(h)}; \qquad |h_a| = \sum_{k=-q}^{c} B_{a.k}^{(h)} 2^k; \qquad \forall B \in \{0,1\} \tag{17}$$

gdzie $c+q+1$ jest długością słowa binarnej reprezentacji wartości h_a.
Operacje logiczne na bitach znaków mają następującą postać:

$$\operatorname{sgn} s_r^{(x)} \cdot \operatorname{sng} h_a = 2\left(\overline{B_{s.r}^{(x)} \oplus B_{s.a}^{(h)}}\right) - 1 \tag{18}$$

Wtedy dla $\forall s \neq 0$ otrzymujemy wartość w formacie mieszanym DM-PCM:

$$s_r^{(x)} h_a = \left[2\left(\overline{B_{s.r}^{(x)} \oplus B_{s.a}^{(h)}}\right) - 1\right] \cdot 2^{l_r} \cdot \sum_{k=-q}^{c} B_{a.k}^{(h)} 2^k \tag{19}$$

oraz w formacie DM-DM:

$$s_r^{(x)} s_a^{(h)} = \left[2\left(\overline{B_{s.r}^{(x)} \oplus B_{s.a}^{(h)}}\right) - 1\right] \cdot 2^{l_r + l_a} \tag{20}$$

Jak łatwo zauważyć, operacje mnożenia sprowadzają się do operacji logicznej negacji sumy modulo 2 oraz zwykłego przesunięcia kodu słowa binarnego w formacie PCM o l_r bitów.

4. STRUKTURY KODERÓW

Struktura kodera ADM jest pokazana na rysunku 5. Koder przetwarza różnicę między sygnałem wejściowym a sygnałem aproksymacji na kod jednobitowy. Realizowane jest to w komparatorze. Następnie różnica jest zamieniana na postać cyfrową w dwupoziomowym układzie kwantowania (SUB) którego sygnałem wyjściowym jest jednobitowy kod ADM. Wartość sygnału aproksymacji jest obliczana w pętli sprzężenia zwrotnego według zasady:

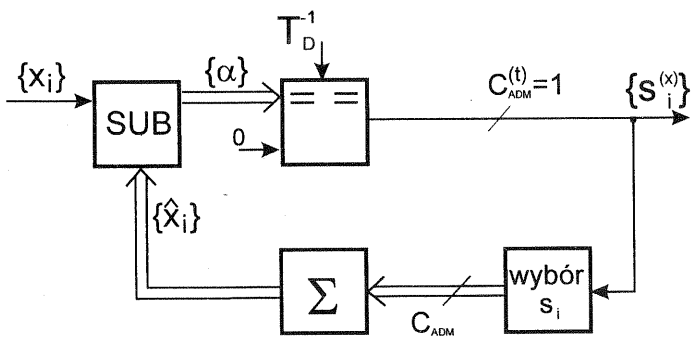
$$\hat{x}_i = \varepsilon \sum_{r=0}^{i} s_r^{(x)} \tag{21}$$

Wybór bieżącego kroku s_i opiera się na analizie kolejno nadchodzących bitów według konkretnego, zaimplementowanego algorytmu, od którego zależy również bitowość c_{ADM}. Należy zauważyć, że bitowość $c_{ADM}^{(t)}$ kodu transmisyjnego ADM wynosi 1 bit, w odróżnieniu od bitowości kroków c_{ADM} wewnątrz struktury kodera i dekodera ADM. Zależności między krokami kwantowania rozważanych w tej pracy rodzajów ADM i DPCM oraz kodami dla transmisji i kodami dla przeprowadzenia operacji matematycznych (operacyjnymi) są następujące:

Kod transmisyjny:		
2^0	2^1	2^2
100	101	111

Kod operacyjny:		
2^0	2^1	2^2
101	110	1100

Z tego widać, że różnica między kodem transmisyjnym i kodem operacyjnym jest równa 1.



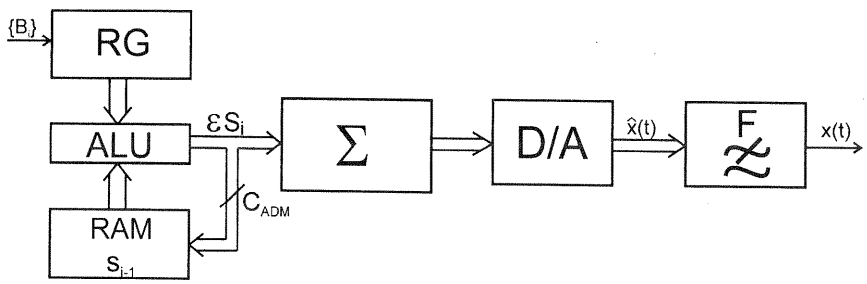
Rys. 4. Struktura cyfrowego kodera ADM

W koderze ADM nie jest konieczne zastosowanie filtra wygładzającego sygnał aproksymacji $\hat{x}(t)$, natomiast użycie analogowego dolnoprzepustowego filtra na wyjściu dekodera staje się niezbędne i ma na celu eliminację górnych składowych widma sygnału dekodera, których nie było w sygnale wejściowym. Tak, jak w większości rodzajów DM, sprzężenie zwrotne kodera może spełniać rolę dekodera.

5. STRUKTURY DEKODERÓW

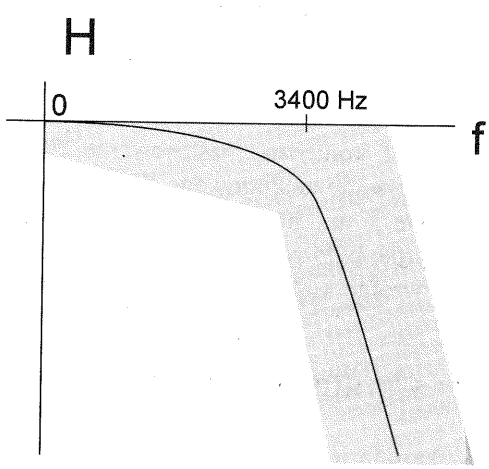
Struktura dekodera pokazana jest na rysunku 5. Jednobitowy sygnał ADM trafia do rejestru przesuwnego RG. Długość rejestru jest zależna od algorytmu konkretnego kodera; np. dla wybranego przez nas kodera ADM długość wynosi 3 bity. Sygnał

z rejestru przesuwnego jest podawany na układ logiczny ALU, który na bazie kombinacji bitów z rejestru przesuwnego oraz wartości ostatniego kroku kwantowania realizuje algorytm kodera i wybiera odpowiedni krok kwantowania. Ostatni krok kwantowania jest przechowywany w pamięci RAM s_{i-1}, a objętość tej pamięci zależy od ilości dostępnych kroków kwantowania. Na przykład dla zbioru kroków $\{-8\varepsilon, -4\varepsilon, -2\varepsilon, -\varepsilon, \varepsilon, 2\varepsilon, 4\varepsilon, 8\varepsilon\}$ objętość pamięci powinna mieć 3 bity. Jest to bitowość kroków c_{ADM} wewnątrz struktury dekodera. Kolejne kroki kwantowania są dodawane w sumatorze Σ. Następnie sygnał cyfrowy jest zamieniany na sygnał analogowy w przetworniku cyfrowo-analogowym D/A.



Rys. 5. Struktura dekodera ADM

Ostatnim etapem odtwarzania sygnału jest filtracja wygładzająca realizowana w dolnoprzepustowym filtrze analogowym F. Granice dopuszczalnej zmiany amplitudowo-częstotliwościowej charakterystyki H(f) tego filtru pokazano na rysunku 6. Zadowalające wygładzanie sygnału mowy w rozważanym koderze ADM uzyskuje się przy wykorzystaniu filtra o nachyleniu zbocza wynoszącym niemniej niż 6 dB/oktawę.



Rys. 6. Charakterystyka amplitudowo-częstotliwościowa analogowego filtru DP wchodzącego w skład dekodera

6. PODSUMOWANIE

Wykonana optymalizacja algorytmów pracy koderów jednobitowej adaptacyjnej modulacji delta (ADM) z krokami kwantowania będącymi krotnością potęgi liczby 2 pozwala na istotne zmniejszenie przepływności strumienia danych w odniesieniu do tradycyjnego PCM przy jednoczesnym zachowaniu dokładności odtwarzania. Wyniki badania algorytmów koderów DPCM wyraźnie pokazują, że kodowanie z wykorzystaniem kroków kwantowania rosnących zgodnie z potęgą liczby 2 daje lepsze rezultaty od kodowania z krokami rosnącymi liniowo. Proponowane struktury kodera i dekodera są proste i ekonomiczne, dlatego mogą być wykorzystane w komunikacji, wieloprocesorowych systemach (na przykład neuronowych) oraz wszędzie tam, gdzie wymagana jest obróbka sygnału (filtracja cyfrowa, analiza korelacyjna, DFT).

7. BIBLIOGRAFIA

1. R. Tadeusiewicz: *Sygnał mowy*. Warszawa, Wydawnictwa Komunikacji i Łączności, 1987.
2. A. V. Oppenheim, R. W. Schafer: *Cyfrowe przetwarzanie sygnałów*. Warszawa, Wydawnictwa Komunikacji i Łączności, 1979.
3. Praca zbiorowa: *Vademecum Teleinformatyka*. Warszawa, IDG Poland S.A., 1999.
4. R. Golański, J. Kołodziej, P. Świrek: *Badanie symulacyjne modulacji różnicowych ze zmiennym krokiem kwantyzacji i nierównomiernym próbkowaniem*. Bydgoszcz, Krajowe Sympozjum Telekomunikacji, 1999, tom B, ss. 298–306
5. W. Pogribny, I. W. Rożankiwski, Z. Drzycimski, J. Różycki: *Częstotliwość dyskretyzacji przetwarzania delta w czasie rzeczywistym sygnałów dwuwymiarowych*. Moskwa-Nowosybirsk, Prace 5 Międzyregionalnej Naukowo-Technicznej Konferencji *Obróbka sygnałów w systemach dwustronnej telefonicznej łączności*, 1995, ss. 49–52. (w jęz. rosyjskim).
6. W. Pogribny, A. Rajs: *Metodyka wykorzystania adaptacyjnej modulacji delta do filtracji cyfrowej*. Bydgoszcz, Krajowe Sympozjum Telekomunikacji, 1996, tom B, ss. 88–94.
7. W. Pogribny, A. Rajs: *Ekonomiczna filtracja cyfrowa z adaptacyjną modulacją delta*. Bydgoszcz, Krajowe Sympozjum Telekomunikacji, 1997, tom B ss. 62–67.
8. W. Pogribny: *Filtering algorithms with delta modulation oriented on the homogeneous computing environments*. Łódź, Proceedings of the 5th Intern. Conf. Mixed Design of Integrated Circuits and Systems MIXDES'98, 1998, pp. 299–302.
9. M. Wiśniewski: *Zarys fonetyki i fonologii współczesnego języka polskiego*. Toruń, Wydawnictwo Uniwersytetu Mikołaja Kopernika, 1997.
10. W. Pogribny, A. Rajs: *Dokładność algorytmów modulacji delta z krokami kwantowania będącymi potęgą liczby 2*. Bydgoszcz, Krajowe Sympozjum Telekomunikacji, 1999, tom B, ss. 44–51.
11. W. Pogribny, A. Rajs: *Research of DM encoders with power 2 quantization steps*. Lviv-Slavsko, Ukraine, Proceedings of the IEEE Intern. Confer. On Modern Problems of Telecommunications, Computer Science and Engineers Training „TCSET'2000'', 2000, pp. 119–120.
12. W. Pogribny, A. Rajs: *Delta coders with the use of quantization steps with multiplication factor equal to power 2*. Ostrava, Czech Republic, Proceedings of the IFAC Workshop on Programmable Devices and Systems PDS 2000, 2000, pp. 103–107.

W. POGRIBNY, A. RAJS

ENCODING AND REPRODUCTION OF RANDOM SIGNALS WITH ECONOMICAL DIFFERENCIAL MODULATIONS

Summary

This paper is devoted to study of differencial aghoritms to random signals encoding. We were considered alghoritms of encoders with multiplication factor equal to power 2, which make possible very simple mathematical operations (code shift only). This is esspecially important to signals procesing. Purpose of this paper is work out and testing accuracy economical encoders algorithms to random signals which based on one-bit adaptative delta modulation (ADM) and differencial PCM (DPCM) in time and frequency domains, and we propose encoder and decoder structures. The exploration were conducted for pink noise and voice signal (according to division on polish phonemes) which was a man, women and child voice. Choice of the algorithms was carried out by means of minimising the mean square error quantization in time and frequency domains, and subjective estimate

Keywords: differencial modulation, voice signal, digitl signal processing.

Dodatek A

Algorytm badanego kodera w postaci kodu języka Pascal:
znaczenie zmiennych: j – liczba naturalna,
liczbad – liczba danych,
PCM[j] – wartość próbki sygnału,
currentstan – wartość $\hat{x}(t)$,
ADM[j] – wartość kroku kwantowania ADM,

Fragment kodu programu:

```
for j: = 1 to liczbad do begin
        if PCM[j] < stan        then wyb_krok_minus6
                                else wyb_krok_plus6;
        current_stan;
                        end
procedure wyb_krok_plus6;
        begin
              ADM[j]: = 1;
              if (ADM[j-1] = 1) and (ADM[j-2] = 1) then ADM[j]: = 2;
              if (ADM[j-1] = 2) then ADM[j]: = 4;
              if (ADM[j-1] = 4) or (ADM[j-1] = 8) then ADM[j]: = 8;
              if (ADM[j-1] =– 8) then ADM[j]: = 4;
              if (ADM[j-1] =– 4) then ADM[j]: = 2;
        end;
procedure wyb_krok_minus6;
        begin
              ADM[j]: =– 1;
              if (ADM[j-1] =– 1) and (ADM[j-2] =– 1) then ADM[j]: =– 2;
              if (ADM[j-1] =– 2) then ADM[j]: =– 4;
              if (ADM[j-1] =– 4) or (ADM[j-1] =– 8) then ADM[j]: =– 8;
              if (ADM[j-1] = 8) then ADM[j]: =– 4;
              if (ADM[j-1] = 4) then ADM[j]: =– 2;
        end;
koniec kodu programu.
```

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2002, 48, z. 1, ss. 27–51

Szeregowanie wieloprocesorowych zadań okresowych dla trzech dedykowanych procesorów

MIROSŁAW GAJER

Katedra Automatyki AGH
Al. Mickiewicza 30, 30-059 Kraków
email:mgajer@ia.agh.edu.pl

Otrzymano 2001.07.05
Autoryzowano 2001.10.09

W ostatnim czasie zaobserwować można było ciągle wzrastającą popularność systemów wieloprocesorowych. Systemy takie są wykorzystywane powszechnie w urządzeniach telekomunikacyjnych, w systemach militarnych, w lotnictwie oraz w nowoczesnych fabrykach, w których zainstalowane zostały roboty przemysłowe. Teoria szeregowania zadań okresowych wywierała i nadal wywiera duży wpływ na rozwój systemów czasu rzeczywistego. Niemniej jednak w klasycznej teorii szeregowania zadań okresowych rozważany jest jedynie przypadek szeregowania zadań jednoprocesorowych – tzn. przyjmuje się, że każde z zadań w danej chwili czasu realizowane jest tylko na jednym procesorze. Mając na uwadze nieustannie wzrastającą popularność rozwiązań wieloprocesorowych, autor niniejszego artykułu opracował własny, oryginalny algorytm szeregowania zadań okresowych, który może zostać zastosowany również w przypadku szeregowania zbioru zadań wieloprocesorowych.

Słowa kluczowe: systemy wieloprocesorowe, obliczenia równoległe, szeregowanie zadań.

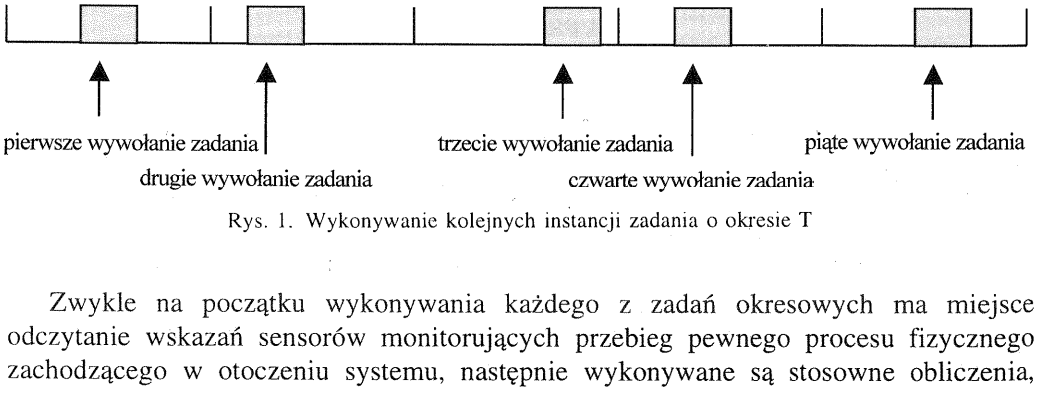
1. WPROWADZENIE

Obecnie systemy czasu rzeczywistego (ang. real-time) stanowią już dosyć dobrze wyodrębnioną i precyzyjnie określoną klasę systemów komputerowych, które odróżniają się w stopniu istotnym od systemów komputerowych ogólnego przeznaczenia [1]. W przypadku systemów czasu rzeczywistego program podzielony jest zwykle na zadania, z których każde realizuje zadane cele systemu [2]. Ponieważ na realizację każdego z zadań nałożone są ograniczenia czasowe, po przekroczeniu których dostarczone przez zadanie wyniki mogą stać się bezużyteczne, niezwykle istotną sprawą jest opracowanie odpowiedniej strategii szeregowania zadań, która będzie w stanie zagwarantować ich terminowe zakończenie. O tym, jak ważne jest to zagadnienie można

przekonać się analizując skutki, jakie może pociągnąć za sobą naruszenie ograniczenia czasowego w systemie o tzw. ostrych ograniczeniach czasowych (ang. hard real-time). W systemach takich przesunięcie wykonywania zdania poza tzw. linię krytyczną (ang. deadline) często grozi wywołaniem awarii bądź katastrofy, a ponadto powstaniem poważnych strat ekonomicznych, a nawet utratą życia ludzi [3]. Z tego względu zagadnienie szeregowania zadań w systemach czasu rzeczywistego wymaga szczególnie gruntownej analizy [4]. Ponieważ w przypadku systemów czasu rzeczywistego o ostrych ograniczeniach czasowych nie można, ze zrozumiałych względów, pozwolić sobie na eksperymentalne sprawdzanie dochowania ograniczeń czasowych w różnych warunkach pracy rozważanego systemu (próby takie mogłyby okazać się zbyt kosztowne lub zbyt niebezpieczne), zatem już na etapie projektowania zachodzi konieczność wykazania, że dany plan zadań jest szeregowalny – tzn. ograniczenia czasowe zostaną dochowane dla każdego z zadań [5].

2. SZEREGOWANIE ZADAŃ W SYSTEMACH CZASU RZECZYWISTEGO

W przypadku systemów czasu rzeczywistego o ostrych ograniczeniach czasowych najpowszechniej występującym typem zadań są tzw. zadania okresowe (ang. periodic tasks). Zadania takie są aktywowane w regularnie następujących po sobie momentach czasu, odległych od siebie o przedział czasu T zwany okresem zadania. Zadanie wykonywane jest przez przedział czasu C, przy czym jego rozpoczęcie może mieć miejsce w dowolnej chwili należącej do danego okresu zadania, pod warunkiem dochowania ograniczenia czasowego D. Z reguły przyjmuje się, że ograniczenie czasowe D równe jest okresowi zadania T [6]. Zatem zadanie musi ulec zakończeniu przed aktywowaniem kolejnej instancji (egzemplarzu, wywołania) tego samego zadania, co zostało zilustrowane na rys. 1.



Rys. 1. Wykonywanie kolejnych instancji zadania o okresie T

Zwykle na początku wykonywania każdego z zadań okresowych ma miejsce odczytanie wskazań sensorów monitorujących przebieg pewnego procesu fizycznego zachodzącego w otoczeniu systemu, następnie wykonywane są stosowne obliczenia,

których celem jest wyznaczenie wartości zmiennych sterujących, po czym wyniki obliczeń wysyłane są jako sygnały sterujące do elementów wykonawczych (aktuatorów). Utrzymanie sterowanego obiektu w obszarze stabilności wymaga cyklicznego powtarzania wyżej wymienionych czynności z pewną minimalną częstotliwością f, której odwrotność równa jest okresowi zadania (T = 1/f).

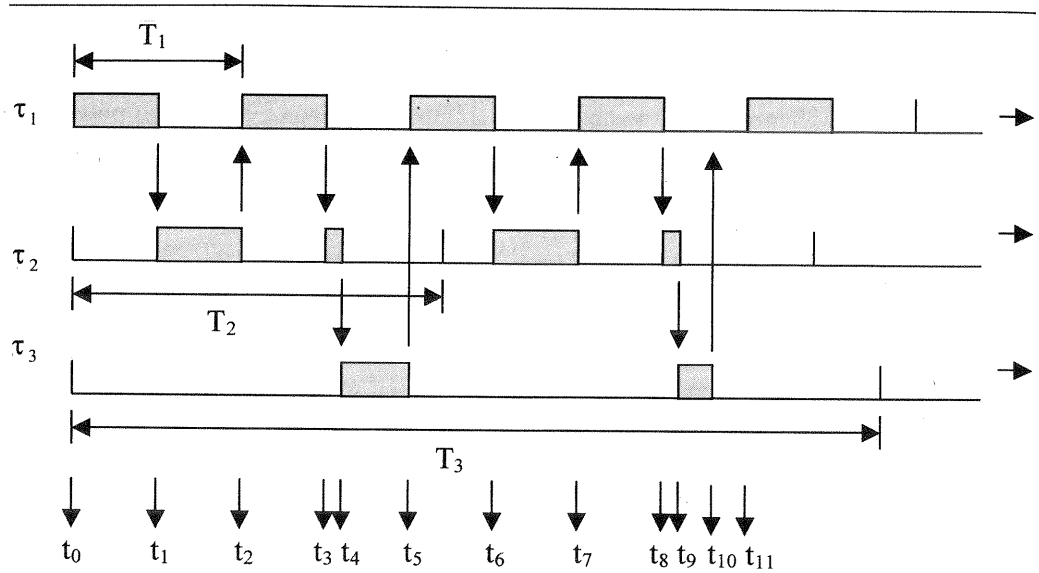
Ponieważ komputerowe systemy czasu rzeczywistego sterują często bardzo złożonymi obiektami technicznymi (np. reaktor jądrowy, system energetyczny, samolot, statek kosmiczny, w pełni zautomatyzowany i zrobotyzowany system produkcji itp.), zatem realizowana przez nie aplikacja składa się z bardzo wielu różnych zadań okresowych o różnych wartościach okresów i czasów wykonania. Często zadania takie są wzajemnie niezależne tzn. nie wymagają wzajemnej komunikacji i synchronizacji. Dzieje się tak zwłaszcza w sytuacji, gdy każde z zadań odpowiedzialne jest za sterowanie odrębnym blokiem bądź podzespołem pewnego większego systemu technicznego [7].

W literaturze przedmiotu opracowano wiele różnych algorytmów szeregowania zadań posiadających ostre ograniczenia czasowe [8]. Dotychczas największą popularnością cieszy się tzw. algorytm szeregowania według monotonicznego tempa RMS (ang. Rate Monotonic Scheduling) [9] (Podana polska nazwa tego algorytmu została zaproponowana w książce [31]). Algorytm ten stosuje się do szeregowania zbioru niezależnych i wywłaszczalnych (ang. preemptive) zadań okresowych, posiadających ostre ograniczenia czasowe. Jego działanie opiera się na kilku prostych zasadach:

1. Zadaniom przydzielane są priorytety, przez program szeregujący zadania (ang. scheduler), na podstawie wielkości okresów tych zadań
2. Wysokości priorytetów są odwrotnie proporcjonalne do wielkości okresów zadań – zatem zadania aktywowane częściej otrzymują wyższy priorytet
3. W danym momencie wykonywane jest zawsze zadanie posiadające najwyższy priorytet
4. Aktywacja zadania o priorytecie wyższym powoduje natychmiastowe wywłaszczenie zadania o priorytecie niższym
5. Wywłaszczone zadanie może zostać wznowione jedynie, gdy żadne inne zadanie o wyższym priorytecie nie znajduje się w stanie gotowości.

Przykład algorytmu RMS zastosowanego do szeregowania zbioru trzech zadań $\{\tau_1, \tau_2, \tau_3\}$ o okresach $\{T_1, T_2, T_3\}$ oraz czasach wykonania $\{C_1, C_2, C_3\}$ zamieszczono na rys. 2.

W chwili t_0 rozpoczyna się wykonywanie pierwszej instancji zadania τ. Ponieważ jest to zadanie o najwyższym priorytecie, wykonywane ono jest w sposób nieprzerwany aż do zakończenia w chwili t_1. Następnie w przedziale czasu od t_1 do t_2 wykonywane jest zadanie τ_2, które w chwili t_2 podlega wywłaszczeniu przez drugą instancję zadania τ. W chwili t_3 wykonywanie zadania τ_2 zostaje wznowione i w chwili t_4 wykonywanie tego zadania zostaje zakończone. W chwili t_4 zostaje również rozpoczęte wykonywanie zadania τ_3, które podlega wywłaszczeniu w chwili t_5, bowiem w chwili t_5 rozpoczyna swoje wykonywanie trzecia instancja zadania τ. W chwili t_6 zadanie τ zostaje zakończone i rozpoczęta zostaje druga instancja zadania τ_2, która w chwili t_7 zostaje wywłaszczona przez czwartą instancję zadania τ. W chwili t_8 zadanie τ zostaje



Rys. 2. Przykładowy plan szeregowania zadań okresowych według algorytmu RMS

zakończone i w związku z tym druga instancja zadania τ_2 zostaje wznowiona. Zadanie τ_2 ulega zakończeniu w chwili t_9 i wtedy również zostaje wznowione wykonywanie zadania τ_3, które zostaje zakończone w chwili t_{10}. Ponieważ w chwili t_{10} żadne z zadań nie znajduje się w stanie gotowości, procesor pozostaje w stanie bezczynnym aż do chwili t_{11}, w której zostaje aktywowana piąta instancja zadania τ.

W pracy [10] zamieszczono dowód twierdzenia, podającego warunek jaki musi spełnić łączny stopień wykorzystania procesora przez zbiór niezależnych, wywłaszczalnych zadań okresowych, aby plan szeregowania tych zadań uzyskany w wyniku zastosowania algorytmu RMS gwarantował dochowanie ograniczenia czasowego dla każdego z zadań.

Zatem niech będzie dany zbiór N niezależnych i wywłaszczalnych zadań okresowych $\{\tau_1, \tau_2, ..., \tau_N\}$ o ograniczeniach czasowych równych okresom tych zadań. Ponadto dane są wartości czasów wykonywania $\{C_2, C_2, ..., C_N\}$ oraz okresów $\{T_1, T_2 ... T_N\}$ rozważanych zadań. Zadanie τ_i wykorzystuje procesor w stopniu C_i/T_i. Jeżeli spełniona jest następująca nierówność

$$\frac{C_1}{T_1}+\frac{C_2}{T_2}+...\frac{C_N}{T_N}\leq N\left(2^{\frac{1}{N}}-1\right) \qquad (1)$$

wówczas ograniczenia czasowe będą zawsze dochowane dla każdego z zadań.

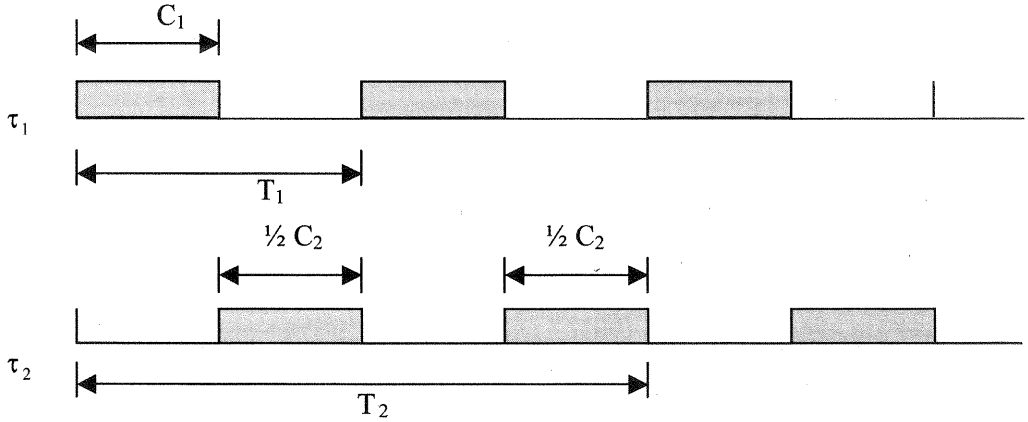
Prawa strona nierówności (1) dla dużych wartości N zmierza szybko do granicy równej ln2 = 0,693. Zatem jeżeli łączny stopień wykorzystania procesora przez szeregowany zbiór zadań nie przekracza 69,3%, rozważany zbiór zadań jest szeregowalny w przypadku zastosowania algorytmu RMS.

Należy zwrócić uwagę, że warunek (1) jest warunkiem wystarczającym na szeregowalność zbioru zadań, nie jest on natomiast warunkiem koniecznym. Zatem jego niespełnienie jeszcze o niczym nie przesądza i badany zbiór zadań może być mimo wszystko szeregowalny. Istotnie, przeprowadzone w pracy [11] badania wykazały, że w rzeczywistych systemach stosujących algorytm RMS średnie wykorzystanie procesora przez szeregowany zbiór zadań wynosi około 88%.

W szczególności nawet zbiór zadań o łącznym 100% wykorzystaniu procesora może być szeregowalny. Dla przykładu zostaną rozważone dwa zadania:

- τ o okresie T_1 = 100ms i czasie wykonywania C_1 = 50ms – zatem $C_1/T_1 = 50\%$
- τ_2 o okresie T_2 = 200ms i czasie wykonywania C_2 = 100ms – zatem $C_2/T_2 = 50\%$

Łączne wykorzystanie procesora przez zadania τ_1 i τ wynosi 100%, a mimo to zadania te są szeregowalne według algorytmu RMS, z zachowaniem ich ograniczeń czasowych, co wynika z przebiegów czasowych realizacji tych zadań przedstawionych na rys. 3.



Rys. 3. Przykład szeregowalności zbioru zadań wykorzystujących w 100% czas pracy procesora

Dla zadanego zbioru zadań, w przypadku niespełnienia nierówności (1) należy sprawdzić, czy czas zakończenia wykonywania każdego z zadań jest mniejszy od jego ograniczenia czasowego. Ponieważ ograniczenia czasowe przyjmuje się jako równe okresom zadań, na początku należy sprawdzić, czy zadanie o najniższym priorytecie zostanie zakończone w ramach swojego okresu. W tym celu należy zastosować następującą procedurę iteracyjną, pozwalająca na znalezienie czasu zakończenia wykonywania zadania o najniższym priorytecie [8].

W kroku pierwszym należy dokonać podstawienia:

$$t_0 = C_1 + C_2 + \ldots + C_N \tag{2}$$

zatem t_0 wyznacza w rozważanym przypadku pierwsze przybliżenie wartości czasu zakończenia wykonywania zadania o najniższym priorytecie.

Następnie należy dokonać podstawienia:

$$t_{k+1} = C_1\left\lceil\frac{t_k}{T_1}\right\rceil + C_2\left\lceil\frac{t_k}{T_2}\right\rceil + \ldots + C_N\left\lceil\frac{t_k}{T_N}\right\rceil \tag{3}$$

($\lceil x\rceil$ – oznacza najmniejsza liczbę całkowitą większą od x)

Procedurę (3) należy powtarzać iteracyjnie, aż do spełnienia warunku:

$$t_{k+1} = t_k \tag{4}$$

Otrzymany czas t_k jest czasem zakończenia wykonywania zadania o najniższym priorytecie. Jeżeli czas ten jest nie większy od okresu tego zadania, wówczas rozważany zbiór zadań jest szeregowalny, w przypadku przeciwnym szeregowalność nie zachodzi. Postępując w ten sposób należy sprawdzić szeregowalność wszystkich zadań występujących w systemie.

Celem ilustracji rozważanych zagadnień zostanie przedstawiony następujący przykład. Niech dany będzie zbiór czterech zadań okresowych o następujących charakterystykach:

- $\tau_1 : T_1 = 120\text{ms}$ i $C_1 = 41\text{ms}$ $\left(C_1/T_1 = 0{,}342\right)$, priorytet = 4 (najwyższy)
- $\tau : T_2 = 180\text{ms}$ i $C_2 = 22\text{ms}$ $\left(C_2/T_2 = 0{,}122\right)$, priorytet = 3
- $\tau_3 : T_3 = 210\text{ms}$ i $C_3 = 64\text{ms}$ $\left(C_3/T_3 = 0{,}305\right)$, priorytet = 2
- $\tau_4 : T_4 = 390\text{ms}$ i $C_4 = 24\text{ms}$ $\left(C_4/T_4 = 0{,}062\right)$, priorytet = 1 (najniższy)

Łączne wykorzystanie procesora przez rozważany zbiór zadań wynosi 0,831. Jest to więcej niż prawa strona nierówności (1), która dla $N = 4$ wynosi $4\left(2^{\frac{1}{4}} - 1\right) = 0{,}757$. Zatem, aby stwierdzić czy rozważany zbiór zadań jest szeregowalny, należy wyznaczyć czas zakończenia zadania o najniższym priorytecie (w tym przypadku zadania τ_4).

Stosując procedurę iteracyjną (3) otrzymuje się kolejno:

$$t_0 = 41 + 22 + 64 + 24 = 151$$

$$t_1 = 41\left\lceil\frac{151}{120}\right\rceil + 22\left\lceil\frac{151}{180}\right\rceil + 64\left\lceil\frac{151}{210}\right\rceil + 24\left\lceil\frac{151}{390}\right\rceil = 41\cdot 2 + 22 + 64 + 24 = 192$$

$$t_2 = 41\left\lceil\frac{192}{120}\right\rceil + 22\left\lceil\frac{192}{180}\right\rceil + 64\left\lceil\frac{192}{210}\right\rceil + 24\left\lceil\frac{192}{390}\right\rceil = 41\cdot 2 + 22\cdot 2 + 64 + 24 = 214$$

$$t_3 = 41\left\lceil\frac{214}{120}\right\rceil + 22\left\lceil\frac{214}{180}\right\rceil + 64\left\lceil\frac{214}{210}\right\rceil + 24\left\lceil\frac{214}{390}\right\rceil = 41\cdot 2 + 22\cdot 2 + 64\cdot 2 + 24 = 278$$

$$t_4 = 41\left\lceil\frac{278}{120}\right\rceil + 22\left\lceil\frac{278}{180}\right\rceil + 64\left\lceil\frac{278}{210}\right\rceil + 24\left\lceil\frac{278}{390}\right\rceil = 41\cdot 3 + 22\cdot 2 + 64\cdot 2 + 24 = 319$$

$$t_5 = 41\left\lceil\frac{319}{120}\right\rceil + 22\left\lceil\frac{319}{180}\right\rceil + 64\left\lceil\frac{319}{210}\right\rceil + 24\left\lceil\frac{319}{390}\right\rceil = 41\cdot 3 + 22\cdot 2 + 64\cdot 2 + 24 = 319$$

Zatem zadanie τ_4 o najniższym priorytecie zostanie zakończone po 319ms. Jest to mniej niż wartość okresu tego zadania równa $T_4 = 390$ms, zatem zadanie o najniższym priorytecie dochowa swojego ograniczenia czasowego.

Postępując w sposób analogiczny można wyliczyć czasy zakończenia pozostałych zadań podlegających procesowi szeregowania, upewniając się, że zadania te dotrzymają swoich ograniczeń czasowych.

Dla zadania τ_3 otrzymuje się:

$$t_0 = 41 + 22 + 64 = 127$$

$$t_1 = 41\left\lceil\frac{127}{120}\right\rceil + 22\left\lceil\frac{127}{180}\right\rceil + 64\left\lceil\frac{127}{210}\right\rceil = 41\cdot 2 + 22 + 64 = 168$$

$$t = 41\left\lceil\frac{168}{120}\right\rceil + 22\left\lceil\frac{168}{180}\right\rceil + 64\left\lceil\frac{168}{210}\right\rceil = 41\cdot 2 + 22 + 64 = 168$$

Zadanie τ zostanie zakończone po upływie 168ms, co stanowi mniej niż okres tego zadania równy $T_3 = 210$ms, czyli zadanie τ dochowa swojego ograniczenia czasowego.

Podobnie dla zadania τ_2 otrzymuje się:

$$t_0 = 41 + 22 = 63$$

$$t_1 = 41\left\lceil\frac{63}{120}\right\rceil + 22\left\lceil\frac{63}{180}\right\rceil = 41 + 22 = 63$$

Zadanie τ_2 zostanie zakończone po upływie 63ms, co stanowi mniej niż okres tego zadania równy $T_3 = 180$ms, a zatem zadanie τ_2 dochowa swojego ograniczenia czasowego.

Z kolei zadanie τ_1 zostanie zakończone po upływie czasu równego 41ms, czyli również przed upływem swojego ograniczenia czasowego 120ms (równego okresowi tego zadania). Stanie się tak dlatego, że zadanie τ_1 jest zadaniem o najwyższym priorytecie, a zatem wykonywane jest zawsze w sposób nieprzerwany (bez wywłaszczeń) od momentu gotowości do zakończenia.

3. BINARYZACJA OKRESÓW ZADAŃ

Binaryzacja okresów zadań jest techniką stosowaną w celu uproszczenia analizy pracy systemów czasu rzeczywistego [12]. Głównym celem procesu binaryzacji okresów zadań jest zmniejszenie tzw. horyzontu czasowego planu szeregowania zadań [13]. Horyzont czasowy definiowany jest jako najmniejsza wspólna wielokrotność okresów zadań i zwykle rośnie wykładniczo wraz z liczbą szeregowanych zadań. Horyzont czasowy stanowi jak gdyby superokres dla wszystkich szeregowanych zadań – jest to zatem okres czasu, po upływie którego cały „scenariusz" pracy systemu powtarza się w sposób cykliczny na nowo.

W procesie binaryzacji okresy wszystkich zadań otrzymują nowe wartości zgodnie z zależnością:

$$T_i' = r \cdot 2^{\left\lfloor \log_2\left(\frac{T_i}{r}\right)\right\rfloor} \tag{5}$$

($\lfloor x \rfloor$ – oznacza największa liczbę całkowitą mniejszą od x)
gdzie r oznacza tzw. okres bazowy, a $i = 1,2,...,N$, gdzie N określa liczbę szeregowanych zadań.

Zatem w wyniku binaryzacji każdy z okresów zadań stanowi pewną całkowitą wielokrotność okresu bazowego r, przy czy dopuszczalne są tylko takie wielokrotności, które stanowią całkowitą potęgę dwójki.

Ponieważ w wyniku binaryzacji okresy zadań ulegają skróceniu, ujemną stroną binaryzacji jest zwiększenie stopnia wykorzystania (obciążenia) procesora przez każde z zadań, gdyż dla $T_i' \le T_i$ zachodzi $\frac{C_i}{T_i'} \ge \frac{C_i}{T_i}$ (T_i – oznacza wartość okresu zadania τ_i przed binaryzacją, natomiast T_i' – oznacza wartość tego okresu po binaryzacji). Zatem po przeprowadzeniu binaryzacji może się okazać – wskutek zwiększenia wykorzystania procesora, że zbiór zadań który był uprzednio szeregowalny, tą szeregowalność utracił.

W tym miejscu należy zastanowić się nad skutkami, jakie może wywrzeć binaryzacja okresów zadań na sterowany system. Otóż, jeżeli sterowany system wymaga zadawania aktualnych wartości zmiennych sterujących z pewną minimalną dopuszczalną częstotliwością f (będącą odwrotnoscią okresu odpowiedniego zadania), to proces binaryzacji spowoduje zwiększenie tej częstotliwości (w skutek zmniejszenia okresu). Zatem sterowanie będzie przebiegać w sposób bardziej dokładny, a margines stabilności sterowanego procesu ulegnie zwiększeniu. Wynika stąd, że binaryzacja okresów zadań jest nie tylko w pełni dopuszczalna, ale nawet pożądana.

Ponieważ binaryzację okresów zadań można przeprowadzić na nieskończenie wiele sposobów (w zależności od arbitralnego wyboru okresu bazowego r) w pracy [14] przedstawiono algorytm wyboru binaryzacji, która pozwala na minimalizację wzrostu stopnia wykorzystania procesora przez szeregowany zbiór zadań, redukując tym samym niebezpieczeństwo utraty szeregowalności.

W celu znalezienia takiej binaryzacji należy dla zbioru N zadań o okresach $\{T_1, T_2, ..., T_N\}$, przy czym $T_1 \le T_2 \le ... \le T_N$, obliczyć N okresów bazowych $\{r_1, r_2, ..., r_N\}$, według wzoru:

$$r_i = \frac{T_i}{2^{\left\lceil \log_2\left(\frac{T_i}{T_1}\right)\right\rceil}} \tag{6}$$

($\lceil x \rceil$ – oznacza najmniejszą liczbę całkowitą większą od x)
dla $i = 1, 2, ..., N$, gdzie N określa liczbę szeregowanych zadań.

Następnie dla każdego z okresów bazowych r_i, $i = 1, 2, ..., N$ należy wyliczyć odpowiadający mu stopień wykorzystania procesora i wybrać ten okres bazowy, dla którego uzyskany stopień wykorzystania procesora jest najmniejszy.

Dla przykładu rozważony zostanie przypadek binaryzacji okresów czterech zadań o okresach $T = \{13, 27, 69, 101\}$ i czasach wykonania $C = \{2, 3, 5, 21\}$. Łączny stopień wykorzystania procesora przez rozważany zbiór zadań okresowych wynosi 0,545.

Dla każdego z okresów należy wyliczyć odpowiadający mu okres bazowy, zgodnie ze wzorem (6)

$$r_1 = \frac{13}{2^{\left\lceil \log_2\left(\frac{13}{13}\right)\right\rceil}} = 13$$

$$r_2 = \frac{27}{2^{\left\lceil \log_2\left(\frac{27}{13}\right)\right\rceil}} = 6{,}75$$

$$r_3 = \frac{69}{2^{\left\lceil \log_2\left(\frac{69}{13}\right)\right\rceil}} = 8{,}625$$

$$r_4 = \frac{101}{2^{\left\lceil \log_2\left(\frac{101}{13}\right)\right\rceil}} = 12{,}625$$

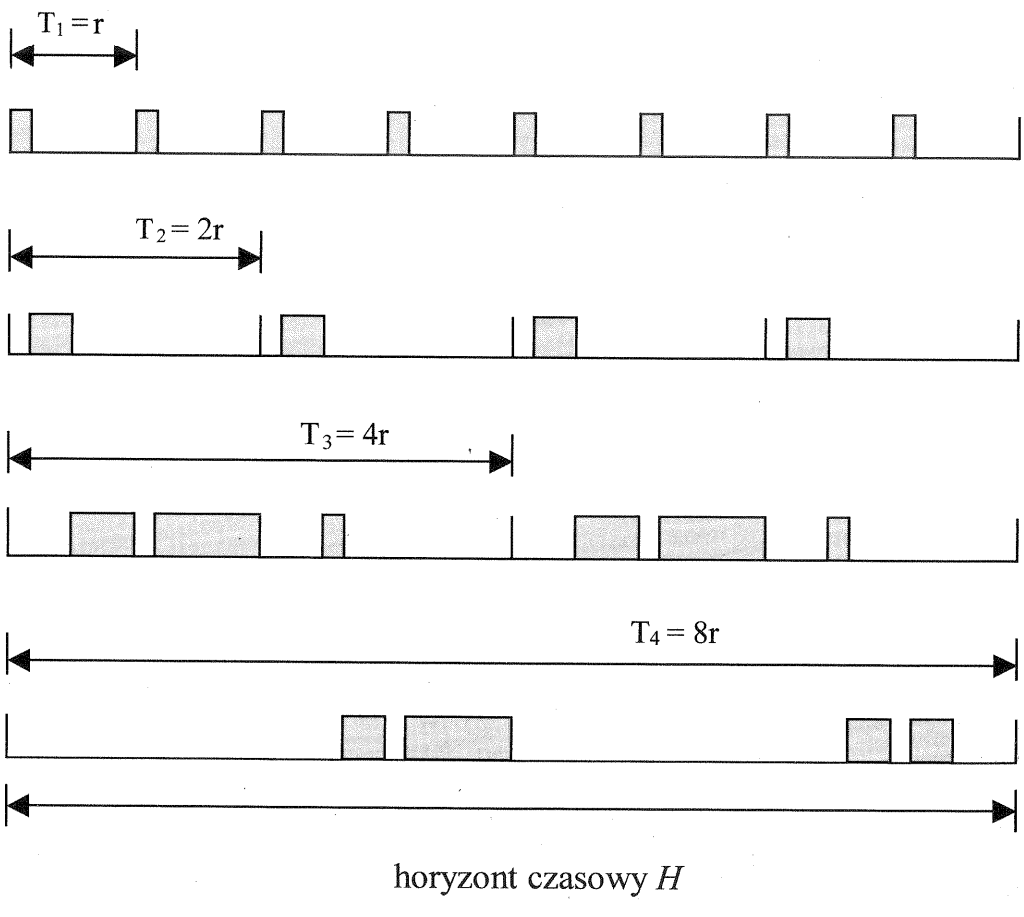
Dla pierwszego z okresów bazowych $r_1 = 13$ otrzymuje się następujący zbiór binarnych okresów zadań $T'_1 = \{13, 26, 52, 52\}$. W tym przypadku stopień wykorzystania procesora przez szeregowany zbiór zadań uległ zwiększeniu do wartości 0,768.

Z kolei dla drugiego z okresów bazowych $r_2 = 6{,}75$ odpowiadający mu zbiór okresów zadań ma postać następującą $T'_2 = \{6.75, 27, 54, 54\}$. W tym przypadku stopień wykorzystania procesora przez szeregowany zbiór zadań uległ zwiększeniu do wartości 0,889. Dla trzeciego z okresów bazowych $r_3 = 8{,}625$ otrzymuje się zbiór okresów zadań $T'_3 = \{8.625, 17.25, 69, 69\}$. W tym przypadku stopień wykorzystania procesora przez szeregowany zbiór zadań uległ zwiększeniu do wartości 0,782.

Dla czwartego z okresów bazowych $r_4 = 12{,}625$ otrzymuje się zbiór okresów zadań $T'_4 = \{12.625, 25.25\,50.5, 101\}$. W tym przypadku stopień wykorzystania procesora przez szeregowany zbiór zadań uległ zwiększeniu do wartości 0,584.

Ponieważ najmniejszy przyrost stopnia wykorzystania procesora uzyskano dla okresu bazowego $r_4 = 12{,}625$, proces binaryzacji okresów zadań należy przeprowadzić w oparciu o ten okres bazowy i jako zbiór okresów binarnych dla szeregowanego zbioru zadań należy przyjąć $T' = \{12.625, 25.25, 50.5, 101\}$. Dla przeprowadzonej w ten sposób binaryzacji przyrost stopnia wykorzystania procesora jest niewielki jeżeli porówna się uzyskaną wartość 0,584, z wartością pierwotną (sprzed binaryzacji) równą 0,545. Ponieważ 0,584 jest mniejsze niż prawa strona nierówności (1) rozważany zbiór zadań jest szeregowalny.

Ponadto warto zwrócić uwagę, że horyzont czasowy szeregowanego zbioru zadań, który stanowi najmniejszą wspólną wielokrotność okresów zadań, uległ zmniejszeniu z wartości 2446119 przed binaryzacją do wartości 101 po binaryzacji. Tak drastyczna



Rys. 4. Przykładowy plan szeregowania zadań o okresach binarnych

redukcja rozmiarów horyzontu czasowego znacznie ułatwia analizę i testowanie pracy systemu, ponieważ cały „scenariusz" jego pracy powtarza się cyklicznie po upływie zaledwie 101 jednostek czasu. W przypadku otrzymanej przed binaryzacją wartości 2446119, przeprowadzenie takiej analizy byłoby praktycznie niemożliwe.

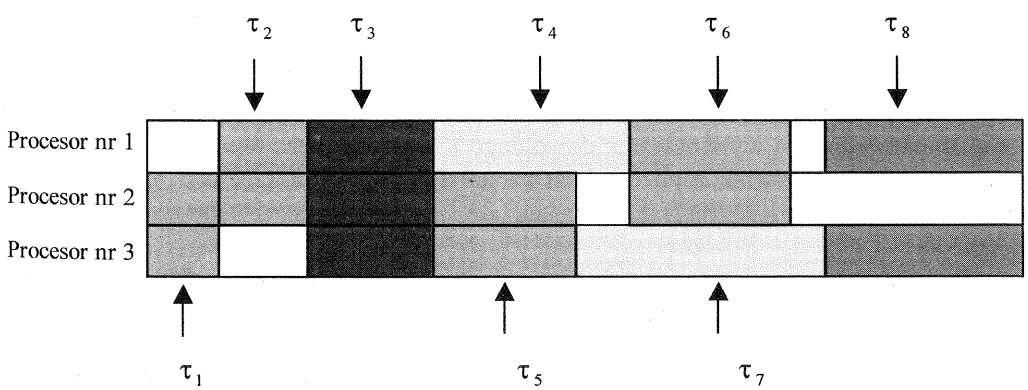
Na rys. 4 przedstawiony został przykładowy plan szeregowania zadań o okresach binarnych.

4. SZEREGOWANIE ZADAŃ WIELOPROCESOROWYCH

W klasycznych modelach szeregowania zadań przyjmuje się zwykle, że każde z zadań wykonywane jest w danej chwili czasu wyłącznie na jednym procesorze [9]. Obecnie wraz ze wzrostem popularności architektur wieloprocesorowych [14], [15], [16], [17], [18] w literaturze przedmiotu pojawia się coraz więcej prac dotyczących szeregowania zadań wieloprocesorowych [19], [20], [21], [22], [23], [24].

Zadanie wieloprocesorowe jest definiowane jako zadanie, które do swojego wykonania potrzebuje jednoczesnej dostępności co najmniej dwóch procesorów [25]. Zbiór procesorów wymaganych do realizacji danych zadań wieloprocesorowych może być określony *a priori* – wówczas mówi się o tzw. przypadku dedykowanych procesorów (ang. dedicated processors), albo też wybór procesorów może być dowolny – taki przypadek nazywa się wariantem z procesorami równoległymi (ang. parallel procesors). W artykule niniejszym uwaga zostanie skupiona na przypadku szeregowania zadań periodycznych dla dedykowanych procesorów.

Zagadnienia związane z szeregowaniem zadań wieloprocesorowych różnią się znacznie, a zwłaszcza ich analiza jest o wiele trudniejsza, niż ma to miejsce w przypadku szeregowania zadań przewidzianych do wykonania na tylko jednym procesorze. W celu dobrego uświadomienia sobie tego faktu przeanalizowany zostanie przykład zamieszczony na rys. 5. W przykładzie tym rozważono szeregowanie zadań aperiodycznych przewidzianych dla trzech dedykowanych procesorów. Szeregowane zadania są reprezentantami różnych typów zadań jedno, dwu i trójprocesorowych.



Rys. 5. Przykład szeregowania zadań wieloprocesorowych

W przykładzie przedstawionym na rys. 5 zadanie τ_1 jest zadaniem dwuprocesorowym, które do swej realizacji potrzebuje jednoczesnej dostępności procesora nr 2 i procesora nr 3. Analogicznie dwuprocesorowe zadanie τ_2 wykonywane jest na procesorze nr 1 i procesorze nr 2. Ponieważ zadania τ_1 i τ_2 do swej realizacji wymagają dostępności tego samego procesora – jest to procesor nr 2 – zadania te nie mogą być jednocześnie wykonywane. Zadania, których zbiory wymaganych procesorów nie są rozłączne, nazywane są zadaniami niekompatybilnymi (ang. incompatible) i zadania takie nie mogą być jednocześnie wykonywane [25]. Z kolei zadanie τ_3 stanowi przykład zadania trójprocesorowego i do jego realizacji wymagana jest dostępność wszystkich trzech procesorów. Zadanie τ_3 jest z oczywistych powodów niekompatybilne z każdym innym zadaniem. Natomiast zadanie jednoprocesorowe τ_4 wykonywane na procesorze nr 1 i zadanie dwuprocesorowe τ_5 wykonywane na procesorach nr 2 i 3, stanowią przykład tzw. zadań kompatybilnych, których zbiory procesorów są rozłączne, a zatem zadania te mogą być realizowane jednocześnie [25]. Analogiczna sytuacja zachodzi w przypadku dwuprocesorowego zadania τ_6 i jednoprocesorowego zadania τ_7, które również stanowią przykład dwóch zadań wzajemnie kompatybilnych. Natomiast zadanie τ_7 nie jest kompatybilne z dwuprocesorowym zadaniem τ_8, ponieważ oba te zadania wymagają jednoczesnej dostępności procesora nr 3.

Widać zatem, że w przypadku szeregowania wieloprocesorowych, niezależnych zadań aperiodycznych, przewidzianych dla dedykowanych procesorów, celem jest znalezienie optymalnego planu szeregowania tych zadań, czyli takiego, który posiadał będzie najkrótszy czas realizacji. W literaturze przedmiotu zagadnienie poszukiwania optymalnego planu szeregowania zadań dla trzech dedykowanych procesorów klasyfikowane jest jako problem typu $P_3|fix_j|C_{\max}$ [26]. W pracy [27] wykazano, że problem $P_3|fix_j|C_{\max}$ należy do klasy problemów NP-trudnych. Zatem w literaturze przedmiotu zaproponowano liczne algorytmy heurystyczne pozwalające na znalezienie w czasie wielomianowym rozwiązań przybliżonych [28], [29], [30].

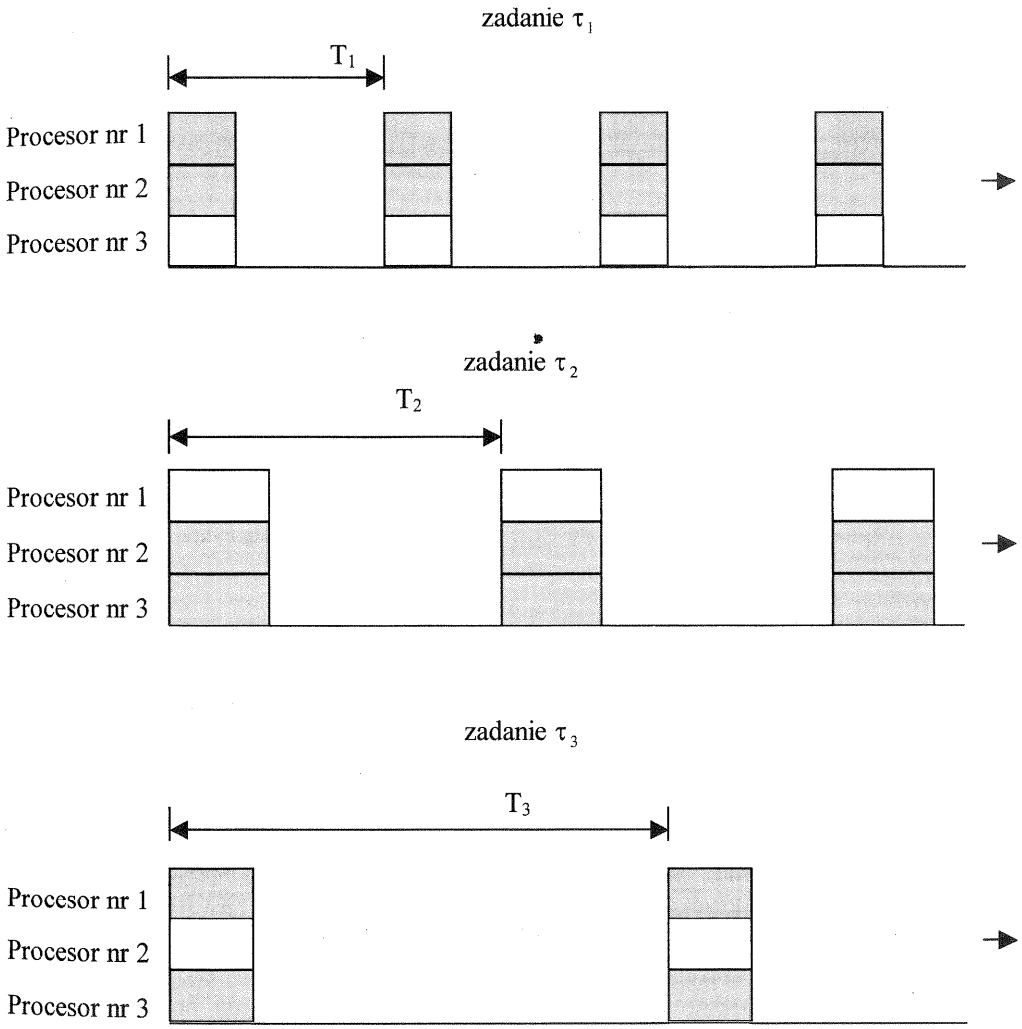
Sprawy komplikują się jeszcze bardziej w sytuacji, gdy rozważane zadania aperiodyczne zostaną przetransformowane do zadań okresowych (np. poprzez przyjęcie jako ich okresu minimalnego przedziału czasu, jaki może upłynąć pomiędzy dwoma kolejnymi przypadkami ich aktywacji [9]) o różnych wartościach okresów.

Przypadek szeregowania okresowych i niezależnych zadań wieloprocesorowych został zilustrowany na rys. 6.

Zamieszczone na rys. 6 dwuprocesorowe zadania τ_1, τ_2 i τ_3 są zadaniami wzajemnie niekompatybilnymi i wykonywanie dowolnego z nich wyklucza możliwość jednoczesnej realizacji jakiegokolwiek z pozostałych zadań. Powstaje problem, jaki należy zastosować algorytm szeregowania takich zadań, aby nie zostały naruszone ich ograniczenia czasowe.

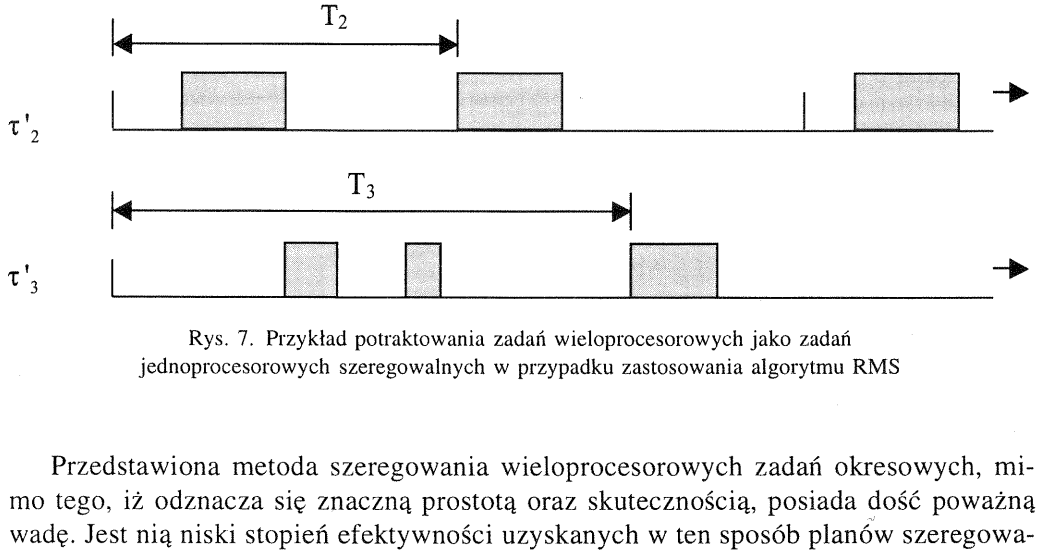
Rozważony zostanie przypadek szeregowania N niezależnych i okresowych zadań wieloprocesorowych przewidzianych dla M procesorów. Najprostszy sposób znalezienia planu szeregowania takich zadań, może polegać na potraktowaniu wszystkich szeregowanych zadań, jako zadań M-procesorowych, czyli takich do realizacji, których potrzebna jest jednoczesna dostępność M procesorów. W przypadku, gdy dane zadanie wymaga

w celu swojej realizacji pewnej liczby procesorów mniejszej niż M, zbiór procesorów zostaje powiększony do M, w ten sposób, że nadmiarowe procesory podczas wykonywania rozważanego zadania będą znajdowały się w stanie pracy jałowej. Ponieważ wszystkie zadania wykonywane są teraz na M procesorach, czyli są to zadania wieloprocesorowe tego samego typu, można je zatem potraktować tak samo, jak zadania jednoprocesorowe, do realizacji których potrzebny jest pewien specyficzny procesor w postaci klastra złożonego z M podjednostek. Na zewnątrz jednak taki klaster widoczny jest, jako pojedynczy procesor, a jego równoległość jest całkowicie przeźroczysta dla użytkownika.



Rys. 6. Przykład zbioru wieloprocesorowych, niezależnych zadań okresowych

Zatem stosując zaproponowana metodę szeregowania zadań wieloprocesorowych do zadań przedstawionych na rys. 6 otrzymuje się trzy zadania trójprocesorowe τ'_1, τ'_2 i τ'_3, które następnie traktowane są jako zadania jednoprocesorowe wykonywane na procesorze, którego wewnętrzna struktura stanowi klaster złożony z trzech podjednostek – fakt ten jest jednakże bezpośrednio niewidoczny dla użytkownika i rozważany klaster traktowany jest jak zwykły pojedynczy procesor. W takim przypadku znalezienie planu szeregowania zadań τ', τ'_2 i τ'_3, które są *de facto* zadaniami jednoprocesorowymi, może polegać na bezpośrednim zastosowaniu teorii szeregowania zadań według monotonicznego tempa RMS. Uzyskany w ten sposób plan szeregowania zadań τ', τ'_2 i τ'_3 został przedstawiony na rys. 7.



Rys. 7. Przykład potraktowania zadań wieloprocesorowych jako zadań jednoprocesorowych szeregowalnych w przypadku zastosowania algorytmu RMS

Przedstawiona metoda szeregowania wieloprocesorowych zadań okresowych, mimo tego, iż odznacza się znaczną prostotą oraz skutecznością, posiada dość poważną wadę. Jest nią niski stopień efektywności uzyskanych w ten sposób planów szeregowania zadań, ponieważ zaproponowana transformacja zadań spowoduje, że przez znaczną część czasu poszczególne procesory będą znajdowały się w stanie jałowym i nie będą w związku z tym wykonywały żadnych użytecznych operacji. Ponadto, uwzględniając warunki szeregowalności planów zadań w przypadku zastosowania algorytmu RMS -nierówność (1) – widać, że liczba zadań, które można w ten sposób uszeregować będzie niewielka i praktycznie potencjalne korzyści jakie daje zastosowanie systemu wieloprocesorowego będą w bardzo nikłym stopniu wykorzystane. Należy zatem opracować algorytm szeregowania zbioru wieloprocesorowych zadań okresowych, który odznaczał się będzie wyższą efektywnością wykorzystania pracy procesorów – procesory przez krótszy czas będą przebywały w bezprodukcyjnym stanie pracy jałowej.

5. ALGORYTM SZEREGOWANIA ZBIORU WIELOPROCESOROWYCH ZADAŃ OKRESOWYCH

Zaproponowany w niniejszym artykule algorytm szeregowania zbioru wieloprocesorowych zadań okresowych składa się z kilku etapów.

W etapie pierwszym należy dokonać binaryzacji okresów wszystkich zadań i znaleźć, zgodnie z procedurą opisaną w punkcie 3, binaryzację, której zastosowanie spowoduje możliwie najmniejsze zwiększenie stopnia zajętości jednostek obliczeniowych.

W etapie drugim, dla każdej z uzyskanej w procesie binaryzacji wartości okresu zadań, należy dokonać zgrupowania zadań wieloprocesorowych, tworząc z nich jedno tzw. zadanie globalne, które przeznaczone jest do jednoczesnego wykonywania przez wszystkie procesory systemu. Ponieważ rozważne zadanie globalne może zostać utworzone z zadań cząstkowych na bardzo wiele sposobów, należy spośród wszystkich możliwych rozwiązań wybrać takie, które charakteryzuje się najkrótszym czasem realizacji. Zadań globalnych tworzonych jest tyle, ile występuje różnych wartości okresów zadań po binaryzacji.

W etapie trzecim należy dokonać szeregowania zadań globalnych wykorzystując w tym celu algorytm RMS. Należy oczywiście sprawdzić, czy uzyskany plan szeregowania zadań globalnych gwarantuje dochowanie ich ograniczeń czasowych.

Celem ilustracji omówionej powyżej metody szeregowania wieloprocesorowych zadań okresowych rozważony zostanie następujący przykład. Niech dany będzie zbiór wieloprocesorowych zadań okresowych przeznaczonych dla trzech dedykowanych procesorów, oznaczonych jako P1, P2 i P3. Charakterystyki szeregowanych zadań zostały zebrane w tab. 1. W kolejnych kolumnach tabeli wymieniono kolejny numer zadania, czas wykonywania, okres zadania, zbiór procesorów koniecznych do jego realizacji, a także stopień wykorzystania jednostek obliczeniowych systemu przez poszczególne zadania.

W przypadku wieloprocesorowego zadania τ o czasie wykonywania C i okresie T, wykonywanego na M procesorach, stopień wykorzystania przez to zadanie jednostek obliczeniowych systemu wyznaczany jest jako $M \cdot \frac{C}{T}$.

Z kolei w tab. 2 zamieszczono charakterystyki rozważanych zadań uzyskane po przeprowadzeniu procesu binaryzacji okresów zadań i wyborze najlepszej binaryzacji. Najlepsza binaryzacja posiada okres bazowy 256ms. W takim przypadku po binaryzacji w systemie występują wyłącznie zadania o okresach 256, 512 i 1024ms.

Wszystkie zadania o okresie 256ms zostały połączone w jedno zadanie globalne τ_1^* w sposób przedstawiony na rys. 8. Kolorem czarnym zaznaczono odcinki czasu, w którym przynajmniej jeden z procesorów znajduje się w stanie pracy jałowej.

Z kolei zadania o okresie 512ms zostały zgrupowane w zadanie globalne τ_2^*, w sposób przedstawiony na rys. 9.

Również zadania o okresie 1024ms zostały połączone w jedno zadanie globalne τ_3^*, jak pokazano na rys. 10.

Tabela 1

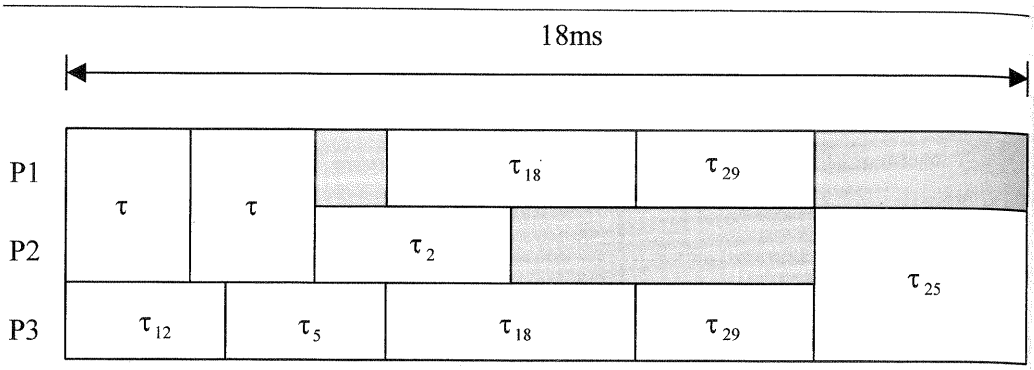
Charakterystyki szeregowanych zadań wieloprocesorowych przed dokonaniem binaryzacji okresów zadań

Zadanie	Czas wykonywania [ms]	Okres zadania [ms]	Zbiór procesorów	Stopień wykorzystania jednostki obliczeniowej
τ_1	4	257	P1, P2	0,03113
τ_2	3	256	P2	0,01172
τ_3	5	512	P1	0,00977
τ_4	13	1024	P1, P2, P3	0,03809
τ_5	6	664	P3	0,00904
τ_6	23	1134	P2	0,02028
τ_7	4	256	P1, P2	0,03125
τ_8	13	1235	P1	0,01053
τ_9	4	609	P2, P3	0,01314
τ_{10}	9	1024	P1, P2	0,01758
τ	3	512	P2, P3	0,01172
τ_{12}	5	289	P3	0,01730
τ_{13}	11	1209	P2	0,00910
τ_{14}	16	1024	P1, P2	0,03125
τ_{15}	4	332	P3	0,01205
τ_{16}	7	709	P2, P3	0,01975
τ_{17}	11	1173	P1, P3	0,01876
τ_{18}	4	377	P1, P3	0,02122
τ_{19}	5	512	P2	0,00977
τ_{20}	12	1203	P2, P3	0,01995
τ_{21}	13	1244	P1	0,01045
τ_{22}	5	654	P3	0,00765
τ_2	4	632	P1, P2, P3	0,01899
τ_{24}	9	1090	P1, P2	0,01651
τ_{25}	3	256	P2, P3	0,02344
τ_{26}	6	576	P2	0,01042
τ_{27}	4	549	P1, P2	0,01457
τ_2	8	1120	P2, P3	0,01429
τ_{29}	2	431	P1, P3	0,00928
τ_{30}	21	1302	P2	0,01613

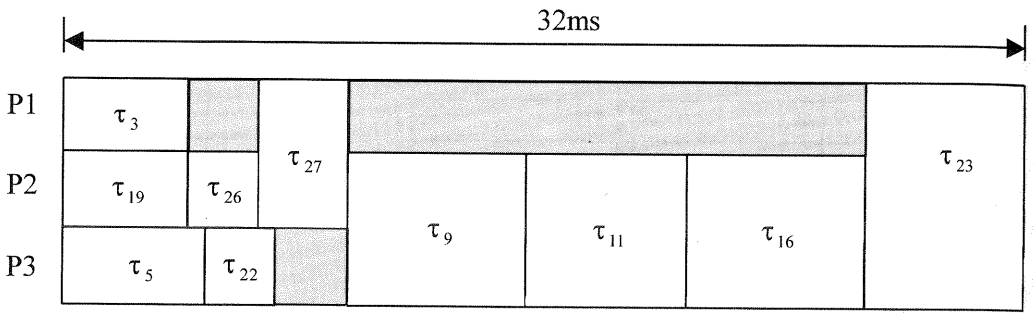
Tabela 2

Charakterystyki szeregowanych zadań wieloprocesorowych po binaryzacji okresów zadań

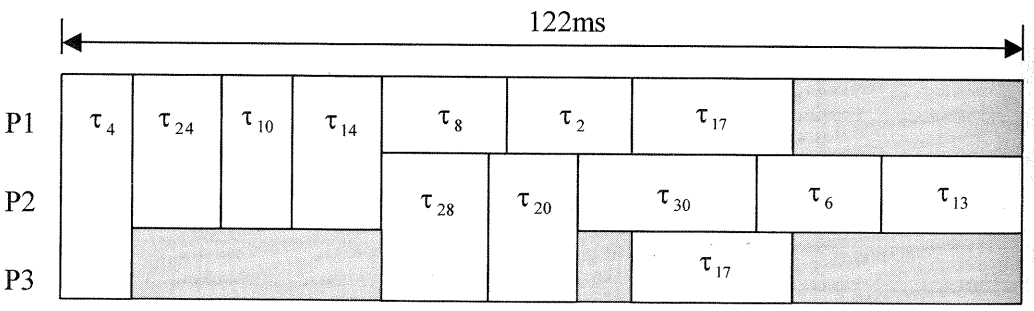
Zadanie	Czas wykonywania [ms]	Okres zadania [ms]	Zbiór procesorów	Stopień wykorzystania jednostki obliczeniowej
τ	4	256	P1, P2	0,03125
τ_2	3	256	P2	0,01172
τ	5	512	P1	0,00976
τ_4	13	1024	P1, P2, P3	0,03809
τ	6	512	P3	0,01172
τ_6	23	1024	P2	0,02246
τ_7	4	256	P1, P2	0,03125
τ	13	1024	P1	0,01269
τ_9	4	512	P2, P3	0,01563
τ_0	9	1024	P1, P2	0,01758
τ	3	512	P2, P3	0,01172
τ_2	5	256	P3	0,01953
τ	11	1024	P2	0,01074
τ_4	16	1024	P1, P2	0,03125
τ	4	256	P3	0,01563
τ_6	7	512	P2, P3	0,02734
τ_7	11	1024	P1, P3	0,02148
τ	4	256	P1, P3	0,03125
τ_9	5	512	P2	0,00976
τ_{20}	12	1024	P2, P3	0,02344
τ_{21}	13	1024	P1	0,01269
τ_{22}	5	512	P3	0,00977
τ_2	4	512	P1, P2, P3	0,03125
τ_{24}	9	1024	P1, P2	0,01758
τ_{25}	3	256	P2, P3	0,02344
τ_{26}	6	512	P2	0,01172
τ_{27}	4	512	P1, P2	0,01563
τ_2	8	1024	P2, P3	0,01563
τ_{29}	2	256	P1, P3	0,01563
τ_{30}	21	1024	P2	0,02051



Rys. 8. Zadanie globalne τ_1^* o okresie 256ms



Rys. 9. Zadanie globalne τ_2^* o okresie 512ms



Rys. 10. Zadanie globalne τ_3^* o okresie 1024ms

Łączny czas wykonywania zadań globalnych wynosi:

- τ_1^* 18ms
- τ_2^* 32ms
- τ_3^* 122ms

Rozważane zadania globalne, do których realizacji potrzebne są trzy procesory mogą zostać potraktowane jako zadania jednoprocesorowe, do których wykonania potrzebny

jest specjalny procesor, w postaci klastra zbudowanego z trzech jednostek obliczeniowych. Jednakże na zewnątrz klaster taki sprawia wrażenie pojedynczego procesora, jego wewnętrzna równoległość jest całkowicie niewidoczna – przeźroczysta dla zewnętrznego użytkownika. W takim przypadku do szeregowania zadań globalnych τ_1^* τ_2^* i τ_3^* może zostać bezpośrednio zastosowana teoria szeregowania zadań RMS.

Można wyliczyć, że zadania globalne wykorzystują jednostkę obliczeniową sytemu (rozważany klaster trójprocesorowy) w stopniu równym:

$$\frac{18}{256}+\frac{32}{512}+\frac{122}{1024} = 0{,}07031+0{,}06250+0{,}11914 = 0{,}25195$$

Jest to mniej niż prawa strona nierówności (1), która dla trzech zadań wynosi $3\left(2^{\frac{1}{3}}-1\right) = 0{,}77976$, a zatem zadania globalne τ_1^*, τ_2^* i τ_3^* są szeregowalne w przypadku zastosowania algorytmu RMS.

W tym miejscu pewnego wyjaśnienia wymaga jeszcze kwestia przypisania priorytetów do zadań globalnych. Otóż w przypadku zastosowania algorytmu RMS, każde z zadań globalnych traktowane jest jak pojedyncze zadanie okresowe, któremu przypisywany jest priorytet odwrotnie proporcjonalny do wielkości okresu tego zadania. W rozważanym przypadku zadaniu globalnemu τ_1^* zostanie przypisany najwyższy priorytet (priorytet = 3) ponieważ zadanie takie posiada najkrótszy okres aktywacji 256 ms. Drugi z kolei priorytet (priorytet = 2) zostanie przypisany zadaniu globalnemu τ_2^* o okresie aktywacji 512 ms. Natomiast zadanie globalne τ_3^* otrzyma najniższy priorytet (priorytet = 1), ponieważ jego okres aktywacji jest najdłuższy i wynosi 1024 ms.

Należy zwrócić uwagę, że zadania cząstkowe wchodzące w skład poszczególnych zadań globalnych zostały uszeregowane w taki sposób, aby ich łączny czas wykonywania był możliwie najkrótszy.

W przypadku niewielkiej liczby zadań cząstkowych wchodzących w skład zadania globalnego (jak miało to miejsce w rozważanym powyżej przypadku) można się jeszcze pokusić o znalezienie rozwiązania optymalnego – tzn. takiego planu szeregowania zadań cząstkowych, który posiada możliwie najkrótszy czas realizacji. Jednakże w ogólnym przypadku jest to zagadnienie bardzo trudne i w praktyce stosowane być mogą jedynie algorytmy heurystyczne, pozwalające na znalezienie pewnego rozwiązania przybliżonego.

Poniżej zamieszczono bardziej szczegółową analizę zagadnienia szeregowania niezależnych zadań wieloprocesorowych dla trzech dedykowanych procesorów.

6. SZEREGOWANIE ZADAŃ DLA TRZECH DEDYKOWANYCH PROCESORÓW

Niech $P = \{1, 2, 3\}$ będzie zbiorem trzech procesorów. Z kolei niech $\mathrm{T} = \{\tau_1 \ldots \tau_N\}$ będzie zbiorem N niezależnych zadań wieloprocesorowych o arbitralnie zadanych czasach wykonywania. Celem swojego wykonania, każde z zadań τ_i, $1 \le i \le N$ wymaga

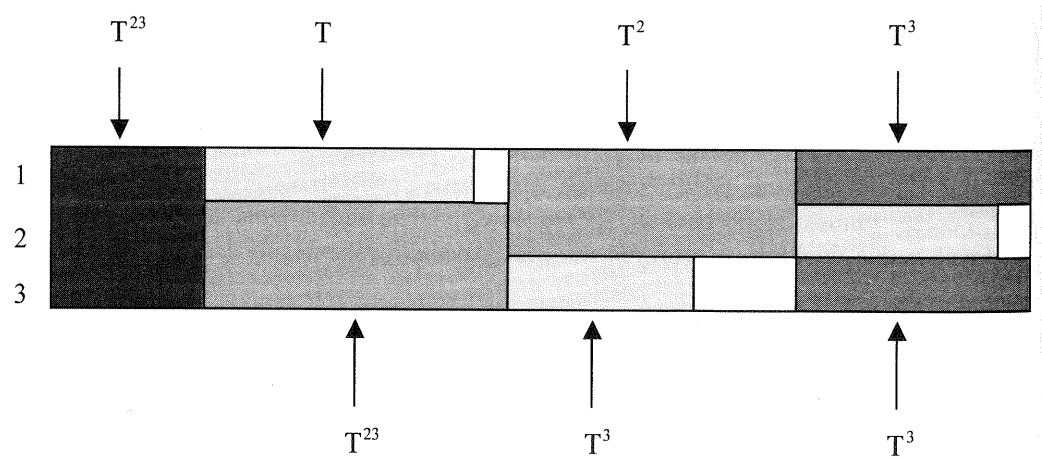
dostępności zdefiniowanego uprzednio niepustego zbioru procesorów przez pewien ciągły przedział czasu $p(\tau_i)$. Zgodnie z wymaganiami odnośnie dostępności procesorów, zbiór zadań T może zostać podzielony na siedem podzbiorów T^1, T^2, T^3, T^{12}, T^{13}, T^{23} i T^{23}. Indeksy górne przy oznaczeniach poszczególnych podzbiorów oznaczają numery procesorów, które konieczne są do wykonania tych zadań. Problem polega na znalezieniu planu szeregowania zbioru zadań T, który posiadać będzie możliwie najkrótszy czas realizacji. Ponieważ każdy z dostępnych procesorów może w danej chwili czasu przetwarzać co najwyżej jedno zadanie, zadania, które do wykonania potrzebują co najmniej jednego wspólnego procesora nie mogą być, z oczywistych względów, równocześnie wykonywane.

Niech symbolem ω oznaczona zostanie funkcja zdefiniowana na podzbiorze zadań T', takim że zawsze zachodzi $T' \subseteq T$. Wówczas $\omega(T')$ odpowiada łącznemu czasowi wykonywania wszystkich zadań wchodzących w skład zbioru T'. Zatem

$$\omega(T') = \sum_{\tau_i \in T'} p(\tau_i).$$

W przypadku szeregowania zbioru T, składającego się z N niezależnych zadań wieloprocesorowych, liczba możliwych do utworzenia planów szeregowania tych zadań jest ograniczona z góry przez liczbę permutacji zadań, czyli przez $N!$. W literaturze przedmiotu wykazano, że znalezienie panu szeregowania niezależnych zadań wieloprocesorowych jest problemem *NP*-trudnym i w praktyce trzeba zadowolić się rozwiązaniami przybliżonymi[32].

We wszystkich pracach z rozważanej dziedziny zadania tego samego typu grupowane były w zbiory, które następnie same tworzyły większe zadania [33]. Takie zadania, składające się z większej liczby zadań tego samego typu były następnie wykonywane w sposób ciągły, tzn. ich realizacja nie była przerywana wykonywaniem żadnego zadania innego typu [34].



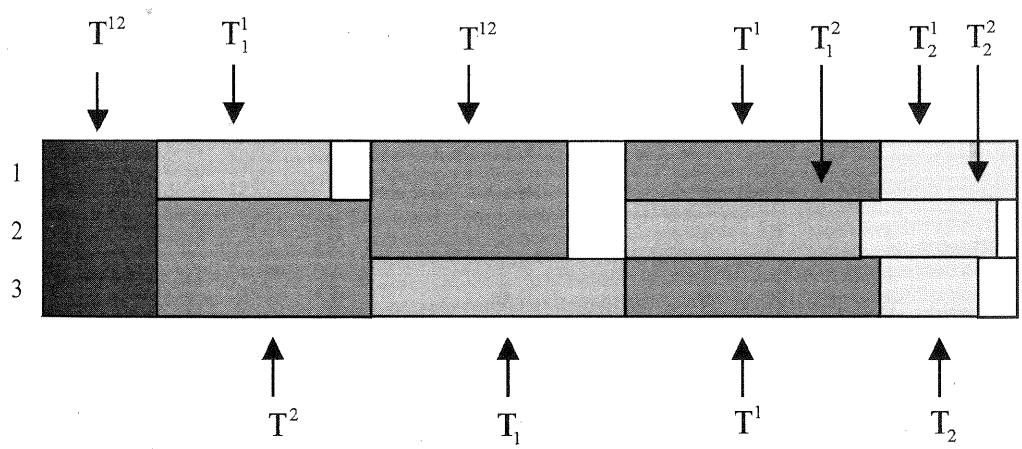
Rys. 11. Przykład planu szeregowania siedmiu podzbiorów zadań wieloprocesorowych przeznaczonych dla trzech dedykowanych procesorów

W przypadku szeregowania zbioru niezależnych zadań wieloprocesorowych T, otrzymuje się siedem podzbiorów zadań tego samego typu: T^1, T^2, T^3, T^{12}, T^{13}, T^{23} i T^{23}. Ich plan szeregowania ma postać jak na rys. 11.

Z rys. 11 widać, że pewne zbiory zadań mogą być wykonywane równocześnie, bowiem dla każdego zbioru zadań dwuprocesorowych T^2, T^3, T^{23}, istnieje odpowiadający im zbiór zadań jednoprocesorowych T, T^2, T^3, które mogą być jednocześnie z nimi wykonywane. Takie pary zbiorów zadań (T, T^{23}), (T^2, T^{13}) oraz (T^3, T^{12}) nazywa się wzajemnie kompatybilnymi (ang. compatible) [25].

Jeżeli czasy realizacji zbiorów zadań dwuprocesorowych są nie mniejsze od czasów realizacji kompatybilnych do nich zbiorów zadań wieloprocesorowych, jak ma to miejsce w przykładzie pokazanym na rys. 11, wówczas zachodzi $\omega(T^1) \leq \omega(T^{23})$, $\omega(T^2) \leq \omega(T^{13})$ oraz $\omega(T^3) \leq (T^{12})$. W takim przypadku, zamieszczony na rys. 11 plan szeregowania zadań wieloprocesorowych dla trzech dedykowanych procesorów jest optymalny, w tym sensie, że posiada najkrótszy czas realizacji.

W przypadku przeciwnym, tzn. gdy $\omega(T^1) > \omega(T^{23})$, $\omega(T^2) > \omega(T^{13})$ i $\omega(T^3) > \omega(T^{12})$ rozwiązanie optymalne nie jest już tak łatwe do znalezienia. W pracy [25] zaproponowano dokonanie podziału (oczywiście w sytuacji, gdy jest to możliwe) każdego zbiorów zadań jednoprocesorowych $T^i, i \in \{1,2,3\}$ na dwa rozłączne podzbiory T_1^i i T_2^i, takie że $T_1^i \cap T_2^i = 0$ i $T_1^i \cup T_2^i = T^i$. Elementy podzbioru T_1^i powinny zostać wybrane w taki sposób, aby bezwzględna wartość różnicy czasów wykonania kompatybilnych zbiorów zadań jedno i dwuprocesorowych $|\omega(T_1^{jk}) - \omega(T_1^i)|$ osiągnęła minimalną wartość dla każdego $i \in \{1,2,3\}$.

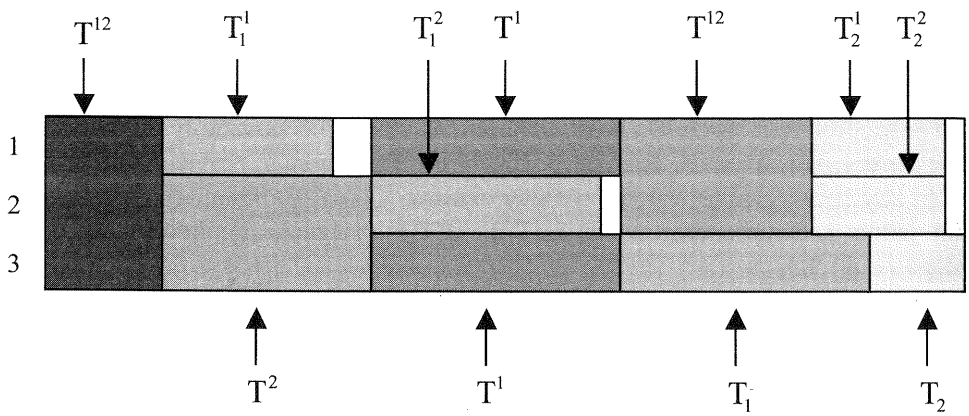


Rys. 12. Przykład planu szeregowania zadań, w którym zbiory zadań T_1^2 i T_2^2 są wykonywane kolejno po sobie

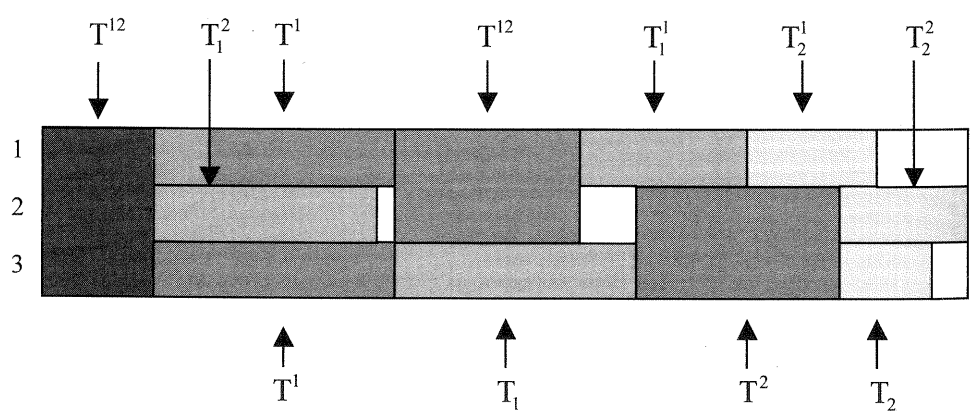
Po dokonaniu podziału zbiorów zadań jednoprocesorowych na dwa rozłączne podzbiory otrzymuje się w sumie dziesięć zbiorów zadań: T_1^1, T_2^1, T_1^2, T_2^2, T_1, T_2, T^{12}, T^1, T^2 i T^{12}. Następnie spośród wszystkich możliwych planów szeregowania powyższych

zbiorów zadań wybiera się ten, który posiada najkrótszy czas realizacji. Okazuje się, że spośród wszystkich możliwych planów szeregowania rozważanych zbiorów zadań wieloprocesorowych tylko trzy są istotne, tzn. że wśród nich należy poszukiwać planu szeregowania o najkrótszym czasie realizacji. Takie trzy przykładowe plany szeregowania zostały zamieszczone na rys. 12, 13 i 14. Z zamieszczonych rys. 12, 13 i 14 widać, że plan posiadający najkrótszy czas realizacji musi być jednym z planów zamieszczonych na tych rysunkach – wszystkie inne możliwe plany szeregowania rozważanych zbiorów zadań posiadają dłuższe czasy realizacji

Spośród przedstawionych na rys. 12, 13 i 14 planów szeregowania zadań, plan przedstawiony na rys. 13, w którym zadania T_1 i T_2 są wykonywane kolejno po sobie posiada najkrótszy czas realizacji i w związku z tym plan ten należy przyjąć jako rozwiązanie końcowe zagadnienia szeregowania zadań wieloprocesorowych dla trzech dedykowanych procesorów.



Rys. 13. Przykład planu szeregowania zadań, w którym zbiory zadań T_1 i T_2 są wykonywane kolejno po sobie



Rys. 14. Przykład planu szeregowania zadań, w którym zbiory zadań T_1^1 i T_2^1 są wykonywane kolejno po sobie

7. ZAKOŃCZENIE

W artykule zaprezentowano sposób działania algorytmu szeregowania wieloprocesorowych zadań okresowych. Pokazano, w jaki sposób połączenie techniki binaryzacji okresów zadań z dobrze znanym z literatury przedmiotu algorytmem szeregowania zadań okresowych RMS, pozwala na rozszerzenie teorii szeregowania zadań okresowych również na przypadek zadań wieloprocesorowych. Autorowi nie są znane z literatury przedmiotu żadne inne prace, w których podjęto by próbę rozszerzenia metody szeregowania zadań okresowych przy użyciu algorytmu RMS na przypadek zadań wieloprocesorowych. W literaturze przedmiotu, jeżeli przypadek zadań wieloprocesorowych jest w ogóle rozważany, to badanie ich szeregowalności rozciąga się jedynie na analizę zbioru zadań aperiodycznych. W świetle powyższych uwag propozycja przedstawiona przez autora niniejszego artykułu, polegająca na zastosowaniu algorytmu RMS oraz techniki binaryzacji okresów zadań do szeregowania wieloprocesorowych zadań periodycznych, stanowi nowatorski wkład do teorii szeregowania zadań w systemach czasu rzeczywistego.

Biorąc pod uwagę nieustannie wzrastającą popularność rozwiązań wieloprocesorowych uzyskane wyniki badań mogą posiadać duże znaczenie praktyczne, przyczyniając się do bardziej efektywnego i racjonalnego wykorzystania jednostek obliczeniowych wieloprocesorowych systemów czasu rzeczywistego.

8. BIBLIOGRAFIA

1. T. Szmuc, G. Motet: *Specyfikacja i projektowanie oprogramowania czasu rzeczywistego*. CCATIE, Kraków, 1998
2. K. G. Shin, P. Ramanathan: *Real-time computing: A new discipline of computer science and engineering*. Proceedings of the IEEE, vol. 82, no. 1, January 1994, pp. 6-24
3. J. H. Lala, R. E. Harper: *Architectural principles for safety-critical real-time applications*. Proceedings of the IEEE, vol. 82, no. 1, January 1994, pp. 25-41
4. J. W. S. Liu, W. K. Shih, K. J. Lin, R. Bettati: *Imprecise Computations*. Proceedings of the IEEE, vol. 82, no. 1, January 1994, pp. 83-94
5. T. Stoyenko, P. Baker: *Real-time schedulability-analyzable mechanisms in Ada9X*. Proceedings of the IEEE, vol. 82, no. 1, January 1994, pp. 95-107
6. K. Ramamritham, J. A. Stankovic: *Scheduling algorithms and operating systems support for real-time systems*. Proceedings of the IEEE, vol. 82, no. 1, January 1994, pp. 55-67
7. N. Suri, M. M. Hugue, C. J. Walter: *Synchronization issues in real-time systems*. Proceedings of the IEEE, vol. 82, no. 1, January 1994, pp. 41-53
8. L. Sha, R. Rajkumar, S. S. Sathaye: *Generalized rate-monotonic scheduling theory: A framework for developing real-time systems*. Proceedings of the IEEE, vol. 82, no. 1, January 1994, pp. 68-82
9. A. Zalewski: *What every engineer needs to know about rate-monotonic scheduling: A tutorial*. Department of Computer Science, The University of Texas of the Permian Basin, Odessa, TX, 1995, pp. 321-335
10. L. Liu, J. W. Layland: *Scheduling algorithms for multiprogramming in a hard real time environment*. Journal on Assoc. Comput. Mach., vol. 20, no. 1, 1973, pp. 46-61
11. D. Gafford: *Rate monotonic scheduling*. IEEE Micro, June 1991

12. J. Czajka, R. Nawrocki: *Szeregowanie zadań o okresach binarnych w systemach silnie uwarunkowanych czasowo.* I Krajowa Konferencja Metody i systemy komputerowe w badaniach naukowych i projektowaniu inżynierskim, Kraków, 1997, ss. 669-676
13. J. Nawrocki, A. Czajka: *Binaryzacja okresów zadań cyklicznych.* VII Konferencja Systemy Czasu Rzeczywistego, Kraków, 2000, ss. 41-51
14. S. D. Haynes, J. Stone, P. Y. K. Cheung, W. Luk: *Video image processing with the sonic architecture.* IEEE Computer, April 2000, pp. 50-57
15. K. H. Kane: *APIs for real-time distributed object programming.* IEEE Computer, June 2000, pp. 72-80
16. G. Bollella, J. Gosling: *The real-time specification for Java.* IEEE Computer, June 2000, pp. 47-54
17. C. Schmidt, F. Kuhns: *An overview of the real-time CORBA specification.* IEEE Computer, June 2000, pp. 56-63
18. B.Chiechi: *Data flow design dictates useable multiprocessor performance on VME bus.* No. 3, 1997, pp. 11-16
19. W. Hsueh, K. J. Lin: *Scheduling real-time systems with end-to-end timing constraints using the distributed pinwheel model.* IEEE Transactions on Computers, vol. 50, no. 1, January, 2001, pp. 51-67
20. J. Verriet: *Scheduling interval-ordered tasks with non-uniform deadlines subject to non-zero communication delays.* Parallel Computing 25, 1999, pp. 3-21
21. *Munier*: Approximation algorithms for scheduling trees with general communication delays. Parallel Computing 25, 1999, pp. 41-48
22. R. H. Möhring, M. W. Schäffter: *Scheduling series-parallel orders subject to 0/1-communication delays.* Parallel Computing 25, 1999, pp. 23-40
23. V. Boeres, E. F. Rebello: *A versatile cost modelling approach for multicomputer task scheduling.* Parallel Computing 25, 1999, pp. 63-86
24. J. Błażewicz, M. Drozdowski, M. Markiewicz: *Divisible task scheduling – Concept and verification.* Parallel Computing 25, 1999, pp. 87-98
25. K. Amoura, E. Bampis, Y. Manoussakis, Z. Tuza: *A comparison of heuristics for scheduling multiprocessor tasks on three dedicated processors.* Parallel Computing 25, 1999, pp. 49-61
26. R. L. Graham, E. L. Lawler, J. K. Lenstra, H. G. R. Rinnooy Kan: *Optimization and approximation in deterministic scheduling: A survay.* Ann. Disc. Math. 5 (1979), pp. 287-326
27. J. Błażewicz, P. Dell'Olmo, M. Drozdowski, M. G. Speranza: *Scheduling multiprocessor tasks on three dedicated processors.* Information Processing Letters 41 (1992), pp. 275-280
28. M. X. Goemans: *An approximation algorithm for scheduling on three dedicated processors.* Discrete Applied Mathematics 61 (1995), pp. 49-59
29. L. Bianco, P. Dell'Olmo, M. G. Speranza: *Nonpreemptive scheduling of independent tasks with prespecified processor allocation.* Naval Research Logistics 41 (1994), pp. 959-971
30. G. Bozoki, J. P. Richard: *A branch-and-bound algorithm for the continuous-process task shop scheduling problem.* AIIE Trans. 2 (1970), pp. 246-252
31. S. Tanenbaum: *Rozproszone systemy operacyjne.* Warszawa, Wydawnictwo Naukowe PWN, 1997, p. 284
32. J. Błażewicz, P. Dell'Olmo, M. Drozdowski, M. G. Speranza: *Corrigendum to scheduling multiprocessor tasks on three dedicated processors.* Information Processing Letters 49 (1994), pp. 269-270
33. A.K. Amoura: *Problémes d'Ordonnancement avec communications dans les systèm multiprocesseurs.* Ph. D. thesis, Université Paris-XI, Orsay, france, 1998
34. K. Amoura, E. Bampis, Y. Manoussakis, Z. Tuza: *Scheduling multiprocessor dedicated tasks on three processors.* Technical report 1089, LRI, Université Paris-Sud, Orsay, 1996

M. GAJER

MULTIPROCESSOR PERIODIC TASKS SCHEDULING FOR THREE DEDICATED PROCESSORS

Summary

Real-time systems are getting more and more popular. They are used among other to control telecommunication systems, defence systems, avionics, and modern factories with industrial robots. Rate-monotonic scheduling theory has had large impact on the development of real-time systems. Nevertheless in classical rate-monotonic scheduling theory it is assumed that each task is processed on one processor at a time. Motivated by the advance of parallel computing systems this author developed a novel method of implementation of rate-monotonic scheduling theory to the case of periodic multiprocessor tasks.

Keywords: multiprocessor systems, parallel computing, task scheduling

W o
gdy każd
struktur
innymi
firmy La
Strul
poszczeg
iloczynó
w znacza
układy.

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2002, 48, z. 1, ss. 53–66

Synteza logiczna dla struktur typu PAL wykorzystująca bufory wyjściowe

DARIUSZ KANIA

Instytut Elektroniki, Politechnika Śląska
ul. Akademicka 16, 44-101 Gliwice
email: kdariusz@zeus.polsl.gliwice.pl

Otrzymano 2001.04.05
Autoryzowano 2001.09.27

Struktura typu PAL stanowi jądro wielu układów CPLD. Jednym z podstawowych problemów syntezy jest problem podziału układu na odpowiednie części, realizowalne w pojedynczych blokach logicznych typu PAL zawierających określoną liczbę iloczynów. W artykule przedstawiono metodę syntezy wykorzystującą trójstanowe bufory wyjściowe, stanowiące dodatkowe zasoby wewnętrzne bloków logicznych. Opracowane algorytmy zostały zaimplementowane i wykorzystane do podziału funkcji testowych w celu realizacji na blokach logicznych typu PAL o zadanej liczbie iloczynów. Przeprowadzono również syntezę funkcji testowych dla standardowych układów PAL, a uzyskane wyniki porównano z klasyczną metodą syntezy i z wynikami opublikowanymi w literaturze.

Słowa kluczowe: synteza logiczna, podział, dekompozycja, minimalizacja, PLD

1. WPROWADZENIE

W ostatnich latach na rynku pojawiło się wiele różnych układów CPLD. Podczas gdy każdy z nich ma unikalną strukturę wewnętrzną, ich jądro często stanowi jedna ze struktur bazowych PLE, PLA lub PAL. W grupie układów typu PAL występują między innymi MAX5000, MAX7000, MAX9000 firmy Altera, MACH firmy AMD, pLSI firmy Lattice, XC7000 firmy Xilinx, AVT2500, AVT5000 firmy Atmel itp. [2].

Struktury typu PAL posiadają ściśle określoną liczbę iloczynów dołączonych do poszczególnych komórek wyjściowych. Najczęściej, choć nie jest to zasadą, liczba iloczynów jest stała dla wszystkich komórek. Ta specyficzna cecha struktur typu PAL w znaczący sposób wpływa na proces syntezy układów cyfrowych w oparciu o tego typu układy. Proces minimalizacji, prowadzony w tym przypadku oddzielnie dla poszczegól-

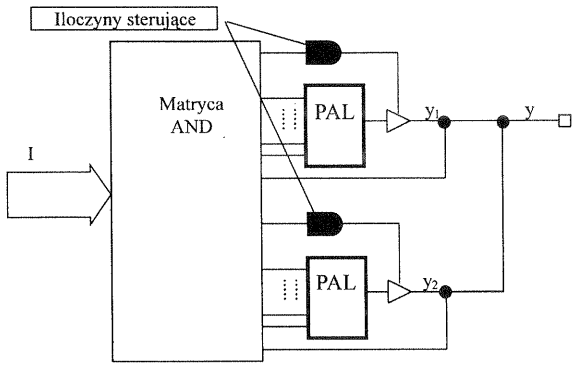
nych funkcji, prowadzi często do dalszego etapu, związanego z ich realizacją na blokach logicznych o niewystarczającej liczbie iloczynów. Istota tego etapu sprowadza się do realizacji funkcji będącej sumą *p*-implikantów na blokach logicznych zawierających *k*-iloczynów, gdy p > k.

Znane z literatury algorytmy dekompozycji najczęściej ukierunkowane są na wielopoziomową realizację układów w strukturach bramkowych lub strukturach typu Look-up Table [20, 21]. Do pierwszej grupy należą przede wszystkim algorytmy syntezy wielopoziomowej oparte na transformacji wyrażeń boolowskich wspomagane procedurami grupowania, leksykograficznego porządkowania zmiennych, programowaniem dynamicznym (MIS-PGA [16], ASYL [1], Chortle [7]). Prace dotyczące dekompozycji układów w celu realizacji w strukturach typu Loo-up Table najczęściej bazują jednak na klasycznym modelu dekompozycji wprowadzonym przez Ashenhursta [3] i rozszerzonym przez Curtisa [10]. Klasyczny model dekompozycji oraz praca [22] są między innymi podstawą systemów TRADE [27], FGSyn [17], Demain [8, 18, 19] oraz algorytmu dekompozycji opublikowanego w [24]. Często dekompozycja wspomagana jest procedurami wykorzystującymi Binary Decision Diagram [17, 24]. Algorytmy dekompozycji stanowią istotę procesu syntezy w oparciu o struktury programowalne [1, 6, 11, 12, 13, 23, 25, 26].

Bloki logiczne typu Loo-up Table wykorzystują bazową strukturę PLE i umożliwiają realizację każdej *n*-wejściowej funkcji, gdzie *n* jest liczbą wejść bloku, stąd podstawowym problemem syntezy jest problem podziału wejść. W przypadku bloków typu PLA istotnym problemem jest problem minimalizacji liczby iloczynów stanowiących wspólne zasoby zespołu funkcji [9]. W przypadku bloków logicznych typu PAL znaczne ograniczenia wprowadzają zasoby wewnętrzne bloku i związana z nimi pojemność kodowa [15]. Podział iloczynów, ze względu na z reguły dużą liczbę wejść występujących w tego typu strukturach, jest jednym z podstawowych problemów syntezy.

Najczęściej zawarte w strukturach CPLD bloki logiczne zawierają dodatkowe zasoby logiczne ułatwiające proces podziału. Należą do nich między innymi sprzężenia typu Folded Nand, często zwane ekspanderami, programowalne rozdzielacze (ang. logic allocator), czy też wyjściowe bufory trójstanowe [2]. Sprzętowe zasoby układów programowalnych umożliwiające nierównomierny podział iloczynów dołączonych do poszczególnych komórek wyjściowych (logic allocator w układach MACH, mixed mode w układach pLSI itp.) mimo swoich wielu zalet nie umożliwiają jednak realizacji każdej funkcji w jednym bloku. W tej sytuacji konieczna jest dodatkowa ekspansja liczby iloczynów. Ekspansja taka, możliwa w większości struktur, wykorzystuje ekspandery, ogólnie rzecz biorąc sprzężenia zwrotne z wyjść na matrycę AND. Umożliwia ona realizację każdej funkcji logicznej, lecz prowadzi niestety do układów znacznie „wolniejszych". W tej sytuacji wydaje się, że najlepszym rozwiązaniem jest ekspansja iloczynów wykorzystująca wyjściowe bufory trójstanowe, która prowadzi ostatecznie do dwupoziomowych (jedna warstwa PAL) struktur (rys. 1).

Idea ekspansji przedstawiona na rysunku 1 ściśle wiąże się z podziałem implikantów realizowanych funkcji. W literaturze [5] znane są idee podziału oraz syntezy



Rys. 1. Idea ekspansji iloczynów wykorzystująca wyjściowe bufory trójstanowe

wielopoziomowych układów na strukturach PAL [4, 23]. Celem niniejszego artykułu jest przedstawienie procesu syntezy dwupoziomowych struktur na blokach logicznych typu PAL zawierających wyjściowe bufory trójstanowe. Punktem wyjścia jest zminimalizowana postać poszczególnych funkcji. Założono, że liczba wejść struktur CPLD jest wystarczająca, stąd proponowany model syntezy nie obejmuje problemu podziału wejść. Przedstawiona w artykule synteza logiczna związana jest bezpośrednio z wewnętrzną strukturą bloków logicznych typu PAL, scharakteryzowanych parametrem *k*-liczbą iloczynów występujących w poszczególnych blokach.

2. PODSTAWY TEORETYCZNE

Niech $y = f(i_{n-1}, \ldots, i_1, i_0)$ będzie funkcją przełączającą, która przyjmuje stan aktywny (0 albo 1) dla określonych wektorów $I = (i_{n-1}, \ldots, i_1, i_0)$ tworzących stany I_j, $(j = 0,1, \ldots, 2^n - 1)$. Zminimalizowana postać funkcji może być przedstawiona za pomocą zbioru wektorów $\mathbf{F}^{(0or1)} = \mathbf{A}_y = \{A_{y(l-1)}, \ldots, A_{y1}, A_{y0}\}$, dla których przyjmuje stan aktywny 0 albo 1. Wybrany wektor zbioru $\mathbf{A}_y$ składa się z *n*-elementów $A_y = (a_{n-1}, \ldots, a_1, a_0)$ odpowiadających zmiennym wejściowym $i_{n-1}, \ldots, i_i, i_0$, przy czym każdy element $a_i \in \mathbf{A}_{\mathbf{0,1,-}} = \{0,1,-\}$, $(i = 0,1, \ldots, n-1)$. Każdy wektor zbioru $\mathbf{A}_y$ pokrywa 2^r stanów I_j, tworzących zbiór stanów $\mathbf{I}$ o mocy 2^r ($\overline{\mathbf{I}} = 2^r$), gdzie r jest liczbą elementów $a_i = -$, przy czym $(i = 0,1, \ldots, n-1)$.

Definicja 1:

Dwa wektory $A_{ys} = (a_{(n-1)s}, \ldots, a_{1s}, a_{0s})$ i $A_{yt} = (a_{(n-1)t}, \ldots, a_{1t}, \ldots, a_{0t})$, których elementy $a_{is}, a_{it} \in \mathbf{A}_{\mathbf{0,1,-}} = \{0,1,-\}$ takie, że A_{ys} pokrywa 2^{r_s} stanów tworzących zbiór stanów $\mathbf{I}_s$, a A_{yt} pokrywa 2^{r_t} stanów tworzących zbiór stanów $\mathbf{I}_t$ nazywamy wektorami półpokrywającymi wtedy i tylko wtedy, gdy $\overline{\overline{\mathbf{I}_s \cap \mathbf{I}_t}} = 2^{r_s - 1}$ lub $2^{r_t - 1}$.

np.: $A_{ys} = (-00-1)_{i_4 i_3 i_2 i_1 i_0}$ i $A_{i_4 i_3 i_2 i_1 i_0}$ są wektorami półpokrywającymi.

Niech $\mathbf{X}$ będzie zbiorem stanów X_j, $(j = 0,1, \ldots, 2^{p-1})$ *p*-elementowych $(p < n)$ wektorów $X = (x_{p-1}, \ldots, x_x, x_0)$, w których poszczególne elementy odpowiadają wy-

branym elementom wektorów $A_y = (a_{n-1}, \ldots, a_1, a_0)$, przy czym każdy element $x_i \in \mathbf{X}_{0,1} = \{0,1\}$, $(i = 0,1, \ldots, p-1)$. Istnieje możliwość odpowiedniego uporządkowania elementów wektora $A_y = (a_{n-1}, \ldots, a_1, a_0)$ załóżmy więc, że

$$\bigwedge_{0 \le i \le p-1} x_i \leftrightarrow a_i \quad \begin{matrix} x_i \in \mathbf{X}_{0,1} = \{0,1\} \\ a_i \in \mathbf{A}_{0,1,-} = \{0,1,-\}. \end{matrix}$$

Takie uporządkowanie powoduje, że dowolny i-ty element wektora $A_y = (a_{n-1}, \ldots, a_1, a_0)$ i $X = (x_{p-1}, \ldots, x_1, x_0)$, gdzie $(i = 0,1, \ldots, p-1)$ odpowiada tej samej zmiennej wejściowej. Uporządkowane pary $\langle a_i, x_i \rangle$ należą do zbioru par, oznaczonego przez $\mathbf{A}$, będącego iloczynem kartezjańskim zbiorów $\mathbf{A}_{0,1,-}$ i $\mathbf{X}_{0,1}$; $(\mathbf{A} = \mathbf{A}_{0,1,-} \times \mathbf{X}_{0,1})$.

Definicja 2:

Dwa wektory $A_y = (a_{n-1}, \ldots, a_1, a_0)$ i $X = (x_{p-1}, \ldots, x_1, x_0)$ takie, że $p \le n$ i $a_i \in \mathbf{A}_{0,1,-} = \{0,1,-\}$, $x_i \in \mathbf{X}_{0,1} = \{0,1\}$ nazywamy wektorami niesprzecznymi wtedy i tylko wtedy, gdy zbiór uporządkowanych p-par $\langle a_i, x_i \rangle$ jest podzbiorem zbioru $\mathbf{A} - \{\langle 0,1 \rangle, \langle 1,0 \rangle\}$, gdzie $\mathbf{A} = \mathbf{A}_{0,1,-} \times \mathbf{X}_{0,1}$.

np.: $A_y = (10-1-)_{i_4 i_3 i_2 i_1 i_0}$ i $X = (011)_{i_3 i_2 i_1}$ są wektorami niesprzecznymi.

Definicja 3:

Dwa wektory $A_y = (a_{n-1}, \ldots, a_1, a_0)$ i $X = (x_{p-1}, \ldots, x_1, x_0)$ takie, że $p \le n$ i $a_i \in \mathbf{A}_{0,1,-} = \{0,1,-\}$, $x_i \in \mathbf{X}_{0,1} = \{0,1\}$ nazywamy wektorami niepseudorównoważnymi wtedy i tylko wtedy, gdy w zbiorze uporządkowanych p-par $\langle a_1, x_1 \rangle$ istnieje para nie należąca do zbioru $\{\langle 0,0 \rangle, \langle 1,1 \rangle\}$.

np.: $A_y = (10-1-)_{i_4 i_3 i_2 i_1 i_0}$ i $X = (100)_{i_3 i_2 i_1}$ są wektorami niepseudorównoważynymi.

Definicja 4:

Wyróżnikiem p-tego rzędu $\overset{X}{\Delta}_{i_{p-1}, \ldots, i_1, i_0}$ nazymy liczbę dziesiętną równą mocy zbioru $\mathbf{A}'_y$, gdzie $\mathbf{A}'_y = \{A'_y \in \mathbf{A}_y$; wektory $A'_y = (a_{n-1}, \ldots, a_1, a_0)$ należące do zbioru $\mathbf{A}'_y$ są niesprzeczne z wektorem $X = (x_{p-1}, \ldots, x_1, x_0)\}$, gdzie $x_i \in \mathbf{X}_{0,1} = \{\mathbf{0,1}\}$, $a_i \in \mathbf{A}_{0,1,-} = \{0,1,-\}$.

Definicja 5:

Wyróżnikiem sprzężonym p-tego rzędu ${}^*\overset{X}{\Delta}_{i_{p-1}, \ldots, i_1, i_0}$ nazywamy liczbę dziesiętną równą mocy zbioru $\mathbf{A}''_y$, gdzie $\mathbf{A}''_y = \{A''_y \in \mathbf{A}_y$; wektory $A''_y = (a_{n-1}, \ldots, a_1, a_0)$ należące do zbioru $\mathbf{A}''_y$ są niepseudorównoważne z wektorem $X = (x_{p-1}, \ldots, x_1, x_0)\}$, gdzie $x_i \in \mathbf{X}_{0,1} = \{0,1\}$, $a_i \in \mathbf{A}_{0,1,-} = \{0,1,-\}$.

Przykład 1:

Jeżeli $\mathbf{A}_y = \begin{bmatrix} 0-011 \\ -0001 \\ 010-1 \\ --010 \\ 1110- \end{bmatrix}_{i_4 i_3 i_2 i_1 i_0}$ i $X = (01)_{i_3 i_0}$

to wyróżnik drugiego rzędu $\Delta^{01}_{i_3, i_0} = \overline{\overline{\mathbf{A}'_y}} = 2$ ponieważ $\mathbf{A}'_y = \begin{Bmatrix} 0-011 \\ -0001 \end{Bmatrix}_{i_4 i_3 i_2 i_1 i_0}$, natomiast wyróżnik sprzężony drugiego rzędu ${}^*\Delta^{01}_{i_3, i_0} = \overline{\overline{\mathbf{A}''_y}} = 4$, ponieważ $\mathbf{A}''_y = \begin{Bmatrix} 0-011 \\ 010-1 \\ --010 \\ 1110- \end{Bmatrix}_{i_4 i_3 i_2 i_1 i_0}$

3. SYNTEZA LOGICZNA DEDYKOWANA DLA STRUKTUR TYPU PAL Z TRÓJSTANOWYMI BUFORAMI WYJŚCIOWYMI

Ogólna koncepcja algorytmu syntezy przedstawiona jest na rys. 2. Proces syntezy zaczyna się od klasycznej dwupoziomowej minimalizacji wykorzystującej algorytm Espresso. Następnie prowadzona jest kolejno synteza poszczególnych zminimalizowanych funkcji, wykorzystująca dwie procedury: rozłączania i podziału wektorów. Wynikiem tych procedur jest podział zbioru wektorów, dla których funkcja przyjmuje stan aktywny, na podzbiory, których moc jest mniejsza od liczby iloczynów zawartych w blokach logicznych typu PAL.



Rys. 2. Koncepcja syntezy dla struktur typu PAL z trójstanowymi buforami wyjściowymi

3a. ROZŁĄCZANIE WEKTORÓW

Dane wejściowe procedury rozłączania wektorów stanowi zbiór wektorów $\mathbf{A}_y = \{A_{y(l-1)}, \ldots, A_{y1}, A_{y0}\}$, dla których funkcja y przyjmuje stan aktywny. Niech wektor $A_{ys} \in \mathbf{A}_y$ równy $A_{ys} = (a_{(n-1)s}, \ldots, a_{1s}, a_{0s})$ pokrywa 2^{r_s} stanów tworzących zbiór stanów $\mathbf{I}_s$, natomiast wektor $A_{yt} \in \mathbf{A}_y$ równy $A_{yt} = (a_{(n-1)t}, \ldots, a_{1t}, a_{0t})$ pokrywa 2^{r_t} stanów tworzących zbiór stanów $\mathbf{I}_t$. Jeżeli dwa wektory $A_{ys} = (a_{(n-1)s}, \ldots, a_{1s}, a_{0s})$ i $A_{yt} = (a_{(n-1)t}, \ldots, a_{1t}, a_{0t})$ są wektorami półpokrywającymi, to okazuje się, że istnieje możliwość modyfikacji jednego z wektorów nie zmieniając zbioru $\mathbf{I}_{s\cup t} = \mathbf{I}_s \cup \mathbf{I}_t$. Wyszukiwanie par wektorów półpokrywających polega na analizie par uporządkowanych $\langle a_{is}, a_{it}\rangle$, gdzie $i = 0,1, \ldots, n-1$. Jeżeli założymy, że wektor A_{ys} zawiera nie mniejszą liczbę elementów a_{is} = „–" niż wektor A_{yt} to wektory A_{ys} i A_{yt} są wektorami półpokrywającymi wtedy, gdy wśród elementów zbioru par uporządkowanych występują pary należące do zbioru $\{\langle 0,0\rangle, \langle 1,1\rangle, \langle -,-\rangle, \langle -,0\rangle, \langle -,1\rangle\}$ i tylko jedna para $\langle a^*_{is}, a^*_{it}\rangle$ należąca do zbioru $\{\langle 1,-\rangle, \langle 0,-\rangle\}$. Modyfikacja, nie zmieniająca zbioru stanów $\mathbf{I}_{s\cup t} = \mathbf{I}_s \cup \mathbf{I}_t$ pokrytych przez wektory A_{ys} i A_{yt}, polega na zamianie elementu a^*_{it} zgodnie z zasadą: jeżeli $a^*_{is} := 1$ to $a^*_{it} := 0$, natomiast jeżeli $a^*_{is} := 0$ to $a^*_{it} := 1$. Jeżeli przykładowo

$$\mathbf{F}^{(1)} = \mathbf{A}_y = \begin{Bmatrix} 0--- \\ --00 \end{Bmatrix}_{i_3 i_2 i_1 i_0} \text{ to po modyfikacji } \mathbf{F}^{(1)} = \mathbf{A}_y = \begin{Bmatrix} 0--- \\ 1-00 \end{Bmatrix}_{i_3 i_2 i_1 i_0}$$

Przedstawiona zasada jest podstawą procedury rozłączania wektorów, która poszukuje wszystkich par wektorów półpokrywających i przekształca jeden z wektorów należących do tej pary. W efekcie tego procesu modyfikowane są wszystkie pary wektorów półpokrywających co prowadzi do uzyskania zbioru wektorów $\mathbf{A}_y$, w którym zminimalizowano liczbę elementów a_i = „–".

Przykład 2:

Algorytm rozłączania wektorów zostanie przedstawiony dla funkcji $y9$ należącej do układu testowego 5xp1.

Po dwupoziomowej minimalizacji uzyskujemy zbiór

$$\mathbf{A}_{y9} = \begin{Bmatrix} ----10- \\ 00--1-0 \\ 1---011 \\ -1--011 \\ 0--01-0 \\ 0-0-1-0 \\ --1101 \end{Bmatrix}_{i_6 i_5 i_4 i_3 i_2 i_1 i_0}$$

Kolejne etapy rozłączania wektorów przedstawione są schematycznie poniżej

$$\left\{\begin{matrix}----10-\\00--110\\1---011\\-1--011\\0--0110\\0-0-110\\--11011\end{matrix}\right\}\rightarrow\left\{\begin{matrix}----10-\\00--110\\1---011\\-1--011\\0--0110\\0-0-110\\--11011\end{matrix}\right\}\rightarrow\left\{\begin{matrix}----10-\\00--110\\1---011\\-1--011\\01-0110\\010-110\\--11011\end{matrix}\right\}\rightarrow\left\{\begin{matrix}----10-\\00--110\\1---011\\01--011\\01-0110\\010-110\\0-11011\end{matrix}\right\}\rightarrow\left\{\begin{matrix}----10-\\00--110\\1---011\\01--011\\01-0110\\010-110\\0011011\end{matrix}\right\}$$

Ostatecznie, po procedurze rozłączania wektorów uzyskujemy zbiór wektorów

$$\mathbf{A}_{y9}=\left\{\begin{matrix}----10-\\00--110\\1---011\\01--011\\01-0110\\0101110\\0011011\end{matrix}\right\}_{i_6i_5i_4i_3i_2i_1i_0}$$

3b. PODZIAŁ WEKTORÓW

Celem procedury podziału wektorów jest podział zbioru wektorów $\mathbf{A}_y$, dla których funkcja przyjmuje stan aktywny na dwa podzbiory $\mathbf{A}'_y$ i $\mathbf{A}''_y$, takie, że moc zbioru $\mathbf{A}'_y$ jest mniejsza lub równa liczbie iloczynów (k) występujących w blokach logicznych typu PAL ($\overline{\overline{\mathbf{A}'_y}} \le k$) natomiast $\overline{\overline{\mathbf{A}''_y}} = \min$.

Jak widać z przykładu 1 zbiór $\mathbf{A}_y$ zawierający wektory, dla których funkcja $y = f(i_{n-1}, \ldots, i_1, i_0)$ przyjmuje stan aktywny można podzielić na dwa podzbiory $\mathbf{A}'_y$ i $\mathbf{A}''_y$ w następujący sposób:

- wybieramy wektor $X = (x_{p-1}, \ldots, x_1, x_0)$ taki, że $p < n$
- zbiór $\mathbf{A}'_y$ tworzymy wybierając ze zbioru $\mathbf{A}_y$ wektory niesprzeczne z założonym wektorem X; $\overline{\overline{\mathbf{A}'_y}} = \Delta^{X}_{i_{p-1},\ldots,i_1,i_0}$
- zbiór $\mathbf{A}''_y$ tworzymy wybierając ze zbioru $\mathbf{A}_y$ wektory niepseudorównoważne z założonym wektorem X; $\overline{\overline{\mathbf{A}''_y}} = {}^{*}\Delta^{X}_{i_{p-1},\ldots,i_1,i_0}$

Ponieważ wybrany wektor należący do zbioru $\mathbf{A}_y$ jest sprzętowo realizowany na jednym iloczynie stąd przedstawiona zasada podziału zbioru $\mathbf{A}_y$ może być natychmiast wykorzystana do realizacji funkcji na elementach typu PAL z wyjściami trójstanowymi (rys. 1).

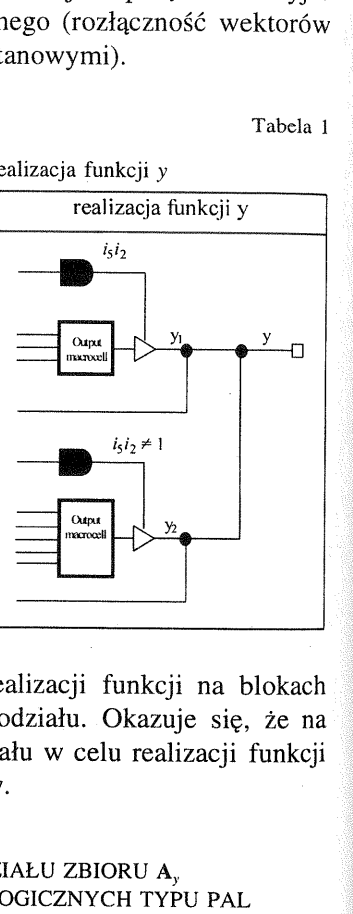
Przykład 3:

Rozważmy funkcję z przykładu 2. Po procedurze rozłączania wektorów otrzymujemy wynik przedstawiony w pierwszej kolumnie tabeli 1. Spróbujmy znaleźć podział implikowany przez wektor $i_5i_2 = 11$. Zbiór wektorów, dla których funkcja y przyjmuje stan aktywny równy 1 można podzielić (zgodnie z przedstawioną zasadą podziału zbioru

$\mathbf{A}_y$) na dwa podzbiory. Pierwszy opisuje funkcję y_1 aktywną dla $i_5 i_2 = 11$ (tab. 1, kolumna 2,) natomiast drugi funkcję y_2 aktywną dla wektorów $i_5 i_2 = 00$ lub 01 lub 10 (tab. 1, kolumna 3). Zbiór wektorów opisujący funkcję y_1 zawiera $\Delta^{11}_{i_5, i_2}$ elementów, natomiast wyróżnik ${}^*\Delta^{11}_{i_5, i_2}$ jest równy liczbie wektorów opisujących funkcję y_2. Implementacja funkcji y po podziale wektorów na poszczególnych blokach typu PAL z wyjściowymi buforami trójstanowymi przedstawiona jest w kolumnie 4 tabeli 1. Opis $i_5 i_2 \neq 11$ oznacza symboliczny opis pozostałej części implikantów nie wchodzących w skład bloku wytwarzającego funkcję y_1, a nie ostateczną realizację. W kolejnym kroku rozpatrywany jest podział bloku wytwarzającego funkcję y_2, (tabela 1, kolumna 4). Ze względu na odpowiedni dobór wektorów podziału możliwe jest połączenie wyjść buforów trójstanowych, nie powodując konfliktu elektrycznego (rozłączność wektorów podziału – funkcji sterujących wyjściowymi buforami trójstanowymi).

Tabela 1

Podział *y.pla* implikowany przez wektor $i_5 i_2 = 11$ oraz realizacja funkcji y

y.pla (5xp1)	y_1 aktywna, gdy $i_5 i_2 = 11$	y_2	realizacja funkcji y
.i 7 .o 10 .ilb I6,I5,I4,I3,I2,I1,I0 .ob y .p 7 ----10- 1 00--110 1 1---011 1 01--011 1 01-0110 1 0101110 1 0011011 1 .e	.i 7 .o 1 .ilb I6,I5,I4,I3,I2,I1,I0 .ob y_1 .p 4 -----0- 1 0--0-10 1 0-01-10 1 .e iloczyn sterujący -1--1-- liczba wektorów = $\Delta^{11}_{i_5 i_2}$	.i 7 .o 1 .ilb I6,I5,I4,I3,I2,I1,I0 .ob y_2 .p 4 -0--10- 1 00--110 1 1---011 1 01--011 1 0011011 1 .e liczba wektorów = ${}^*\Delta^{11}_{i_5 i_2}$	

W przykładzie 3 została pokazana idea podziału i realizacji funkcji na blokach logicznych typu PAL. Wektor $i_5 i_2$ nazwijmy wektorem podziału. Okazuje się, że na podstawie wyróżników można poszukiwać wektorów podziału w celu realizacji funkcji na blokach logicznych o ściśle określonej liczbie iloczynów.

3C. ALGORYTM WYSZUKIWANIA WEKTORA PODZIAŁU ZBIORU $\mathbf{A}_y$ PROWADZĄCY DO REALIZACJI FUNKCJI NA BLOKACH LOGICZNYCH TYPU PAL O ZADANEJ LICZBIE ILOCZYNÓW

Załóżmy, że liczba iloczynów dołączona do poszczególnych komórek wyjściowych jest stała i wynosi k.

Istota algorytmu polega na poszukiwaniu takiego wektora podziału $X = (x_{p-1}, \ldots, x_1, x_0)$, dla którego $\Delta_{i\,p-1,\ldots,i_1,i_0}^{X} \leq k$ i ${}^{*}\Delta_{i\,p-1,\ldots,i_1,i_0}^{X} = \min$. Wektor podziału X wyszukiwany jest na podstawie wyróżników w sposób przedstawiony poniżej.

1. Wyznaczamy wszystkie wyróżniki pierwszego rzędu oraz odpowiadające im wyróżniki sprzężone. Jeżeli wśród wyróżników istnieją takie, których wartość jest mniejsza lub równa k (liczba iloczynów doprowadzona do komórki wyjściowej), to jako wektor podziału wybieramy taki wektor, dla którego odpowiadający mu wyróżnik sprzężony przyjmuje wartość minimalną. Jeżeli wyróżnik sprzężony jest mniejszy lub równy k, to proces podziału jest zakończony, w przeciwnym przypadku poszukujemy identyczną metodą wektora podziału dla podzbioru $\mathbf{A}'_y$ zakładając, że zmienne wyszukiwanego wektora należą również do kolejnego wektora podziału (ponownie punkt 1).
2. Jeżeli wszystkie wyróżniki pierwszego rzędu są większe od k, to wyznaczamy wyróżniki drugiego rzędu, szukając takiego, który jest mniejszy lub równy k, a odpowiadający mu wyróżnik sprzężony przyjmuje wartość minimalną. Jeżeli wyróżnik taki istnieje, to odpowiadający mu wektor jest wektorem podziału, natomiast dla pozostałych wektorów (jeżeli liczba ich jest większa od k) szukamy wektora podziału tak jak w p. 1.
3. Jeżeli wszystkie wyróżniki drugiego rzędu są większe od k, to wektora podziału szukamy wyznaczając wyróżniki 3, 4, itd. rzędu.
4. Algorytm wyszukiwania wektorów podziału kończy się w sytuacji, gdy liczba pozostałych wektorów zbioru $\mathbf{A}_y$ jest mniejsza lub równa k.

Przykład 4:

Załóżmy, że chcemy zrealizować układ testowy 5xpl na blokach logicznych typu PAL zawierających 3 iloczyny. Wykorzystując przedstawiony algorytm można poszukiwać wektorów podziału dla poszczególnych funkcji, po minimalizacji i procedurze rozłączania wektorów. W pierwszym i trzecim wierszu tablicy 2 zawarte są pliki **.pla* opisujące poszczególne funkcje 5xpl, które muszą podlegać podziałowi, ponieważ zminimalizowana postać wymaga wykorzystania liczby iloczynów większej od 3. Drugi i czwarty wiersz tablicy 2 przedstawia schemat podziału dla poszczególnych funkcji (nad strzałkami umieszczone są wektory podziału, w węzłach wartości wyróżników).

Tabela 2

Podział funkcji należących do 5xpl prowadzący do realizacji na blokach logicznych typu PAL, zawierających 3 iloczyny

<table>
<tr>
<td>

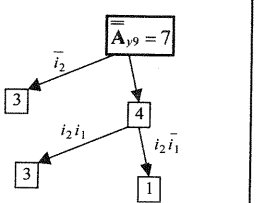
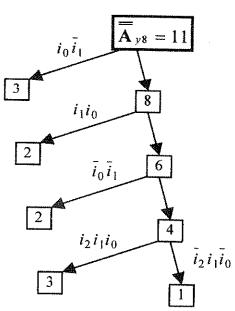
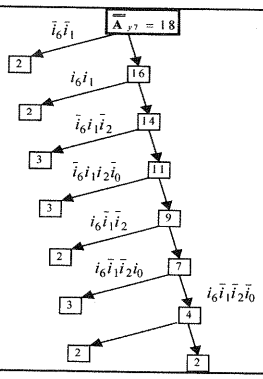
```
.i 7
.o 1
.ilbI6,I5,I4,I3,I2,I1,I0
.ob y9
.p 7
----10- 1
00--110 1
1---011 1
01--011 1
01-0110 1
0101110 1
0011011 1
.e
```

</td>
<td>

```
.i 7
.o 1
.ilbI6,I5,I4,I3,I2,I1,I0
.ob y8
.p 11
----010 1
----101 1
11--001 1
101-001 1
111-100 1
1101100 1
00--110 1
00-0011 1
0001011 1
01-0110 1
0101110 1
.e
```

</td>
<td>

```
.i 7
.o 1
.ilbI6,I5,I4,I3,I2,I1,I0
.ob y7
.p 18
1---010 1
0---001 1
100-001 1
111-000 1
1101000 1
0---100 1
10--100 1
1100100 1
0111010 1
00-0011 1
0001011 1
1---111 1
01--111 1
001111- 1
0--0110 1
0-01110 1
11--101 1
101-101 1
.e
```

</td>
</tr>
<tr>
<td>  </td>
<td>  </td>
<td>  </td>
</tr>
<tr>
<td>

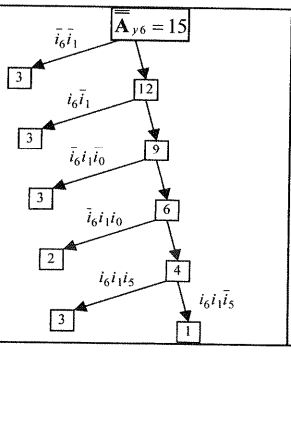
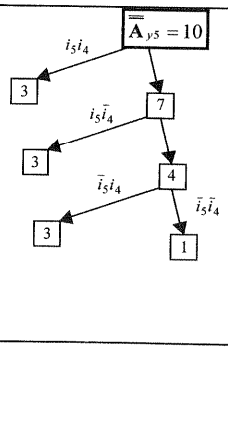
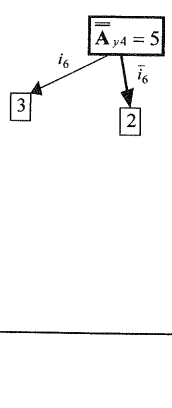
```
.i 6
.o 1
.ilbI6,I5,I4,I3,I1,I0
.ob y6
.p 15
01--01 1
00--10 1
100-0- 1
001101 1
011100 1
110000 1
111-1- 1
11011- 1
01-010 1
010110 1
101-00 1
00-011 1
000111 1
110011 1
101-11 1
.e
```

</td>
<td>

```
.i 5
.o 1
.ilbI6,I5,I4,I3,I0
.ob. y5
.p 10
-00-1 1
-0110 1
-1000 1
00101 1
-1111 1
11101 1
11011 1
10100 1
01100 1
01010 1
.e
```

</td>
<td>

```
.i 4
.o 1
.ilbI6,I5,I4,I3
.ob y4
.p 5
0-10 1
1-11 1
0101 1
1-00 1
1001 1
.e
```

</td>
</tr>
<tr>
<td>  </td>
<td>  </td>
<td>  </td>
</tr>
</table>

4. WYNIKI EKSPERYMENTÓW

Przedstawiony w pracy algorytm został zaimplementowany i stanowi rozszerzenie programu Decomp, przeznaczonego do syntezy układów cyfrowych w strukturach programowalnych. Uzyskane wyniki syntezy dla kilku popularnych funkcji testowych (ang. benchmark) zostały przedstawione w tabeli 3. Podział przeprowadzono dla bloków logicznych typu PAL scharakteryzowanych wartością k (liczbą iloczynów dołączonych do komórki wyjściowej), występujących w strukturach CPLD [2]. Uzyskane wyniki porównano ze strategią klasyczną, w której minimalizacja wykonana została dla każdej funkcji oddzielnie, po czym wykorzystano ekspansję liczby iloczynów za pomocą wewnętrznych sprzężeń zwrotnych.

Tabela 3

Wyniki syntezy (liczba wykorzystanych bloków logicznych/liczba warstw bloków typu PAL) dla wybranych układów testowych (k-liczba iloczynów dołączonych do komórki wyjściowej)

				Metoda klasyczna					Zaproponowana metoda				
				k =					k =				
	i	o	p	4	5	6	7	8	4	5	6	7	8
5xp1	7	10	75	26/3	22/2	18/2	16/2	15/2	25/1	20/1	18/1	17/1	15/1
bw	5	28	87	38/2	30/2	28/1	28/1	28/1	38/1	30/1	28/1	28/1	28/1
rd53	5	3	32	11/2	8/2	6/2	6/2	6/2	10/1	9/1	7/1	7/1	5/1
con1	7	2	9	3/2	2/1	2/1	2/1	2/1	3/1	2/1	2/1	2/1	2/1
misex1	8	7	32	12/2	8/2	7/1	7/1	7/1	12/1	8/1	7/1	7/1	7/1
misex2	25	18	29	19/2	18/1	18/1	18/1	18/1	19/1	18/1	18/1	18/1	18/1
xor5.pla	5	1	16	5/2	4/2	3/2	3/2	3/2	4/1	4/1	4/1	4/1	2/1
b12	15	9	431	19/2	15/2	14/2	11/2	11/2	21/1	16/1	13/1	10/1	10/1
inc	7	9	49	17/2	16/2	13/2	11/2	11/2	18/1	16/1	14/1	11/1	11/1
squar5	5	8	32	11/2	9/2	9/2	9/2	8/1	10/1	9/1	9/1	9/1	8/1
z5xp1	7	10	128	27/3	23/2	19/2	17/2	15/2	27/1	22/1	19/1	17/1	16/1
			Σ	188/24	155/20	137/18	128/18	124/17	187/11	154/11	139/11	130/11	122/11

Analizując uzyskane wyniki można zauważyć, że zaproponowana metoda syntezy jest bezkonkurencyjna jeżeli chodzi o liczbę warstw logicznych uzyskiwanej struktury. Dla przedstawionych układów testowych liczba wykorzystywanych bloków logicznych w dwóch metodach syntezy jest podobna.

Przeprowadzono również syntezę dla struktur PAL20V8, a uzyskane wyniki porównano z wynikami opublikowanymi w [4]. Pierwsza liczba określa liczbę użytych struktur, natomiast druga liczbę poziomów logicznych.

Tabela 4

Wyniki syntezy logicznej wykorzystującej struktury PAL20V8

	Zaproponowana metoda	ASYL [4]
rd53	1/1	1/1
bw	3/1	3/1
inc	3/1	–
misex1	1/1	1/1
misex2	3/1	3/1
squar5	2/1	–
5xp1	2/1	2/2
rd73	4/1	3/2
f51m	2/1	2/2

5. PODSUMOWANIE

Głównym ograniczeniem bloków logicznych typu PAL jest stosunkowo mała liczba iloczynów, co sprawia, że proces ich wykorzystania staje się podstawowym problemem syntezy. Bloki logiczne typu PAL często posiadają trójstanowe bufory wyjściowe. Prezentowana metoda syntezy umożliwia wykorzystanie tych elementów prowadząc ostatecznie do realizacji układów w postaci dwupoziomowych (jedna warstwa PAL) struktur logicznych. Opracowana procedura rozłączania wektorów stanowi przygotowanie do dalszego procesu podziału znacząco zwiększając jego efektywność. Dwa etapy syntezy tzn. rozłączanie i podział wektorów zostały zaimplementowane w prototypowym systemie, a uzyskane wyniki wskazują na konkurencyjność przedstawionej metody syntezy z innymi metodami.

6. BIBLIOGRAFIA

1. P. Abouzeid, B. Babba, M. Crastes, G. Saucier: *Input-Deiven Partitioning Methods and Application to Synthesis on Table-Lookup-based FPGAs.* IEEE Trans on CAD, 12, 1993, No. 7, pp. 913–925
2. AMD, Altera, Lattice Data Book
3. R.L. Ashenhurst: *The decomposition of switching functions.* Proceedings of an International Symposium on the Theory of Switching, April 1957
4. B. Babba, M. Crastes, G. Saucier: *Input driven synthesis on PLDs and PGAs.* The European Conference on Design Automation, Brussels (Belgium), March 1992

5. M. Bolton: *Digital Systems Design with Programmable Logic.* Addison-Wesley Publishing Company, 1990, pp. 133–140
6. R.K. Brayton, G.D. Hachtel, A.L. Sangiovanni-Vincentelli: *Multilevel logic synthesis.* Proc. Of the IEEE, 78(2), February 1990, pp. 264–300
7. S.D. Brown, R.J. Francis, J. Rose, Z.G. Vranesic: *Field Programmable Gate Arrays.* Boston, Kluwer Academic Publishers, 1993, pp. 45–86
8. J.A. Brzozowski, T. Łuba: *Decomposition of Boolean Functions Specified by Cubes.* University of Waterloo Computer Science Department, CS-97-01, January 1997
9. M.J. Ciesielski, S. Yang: *PLADE: A two-stage PLA decomposition.* IEEE Transactions on Computer-Aided Design, 11(8), August 1992, pp. 943–954
10. H.A. Curtis: *The Design of switching Circuits*: New Jersey, D. van Nostrand Company, Inc., 1962
11. F. Dresig, Ph. Lanches, O. Rettig, U.G. Baitiger: *Functional Decomposition for Universal Logic Cells using Substitution.* The European Conference on Design Automation, Brussels (Belgium), March 1992
12. L. Jóźwiak: *General decomposition and its use in digital circuit synthesis.* VLSI Design, 3 (3–4), 1995, pp. 225–248
13. L. Jóźwiak, F. Volf: *An efficient method for decomposition of multiple-output Boolean functions and assigned sequential machines.* The European Conference on Design Automation, Brussels (Belgium), March 1992
14. D. Kania: *Two-level logic synthesis on PALs.* Electronics Letters, 1999, Vol.35, No. 11, pp. 879–880
15. D. Kania: *Coding Capacity of Programmable Transcoder.* Kwartalnik Elektroniki i Telekomunikacji, 1998, T. 44, z. 2, ss. 193–203
16. R. Murgai, Y. Nishizaki, N. Shenay, R.K. Brayton, A. Sangiovanni-Vincentelli: *Logic Synthesis for Programmable Gate Array.* Proc. 27th DAC, June 1990, pp. 620–625
17. Y. Lai, K. Pan, M. Pedram: *FPGA synthesis using function decomposition, Proc.* of the IEEE International Conference on Computer Design, Cambridge, 1994, pp. 30–35
18. T. Łuba, H. Selveraj, A. Kraśniewski: *A new approach to FPGA-based logic synthesis.* Workshop on Design Methodologies for Microelectronics and Signal Processing, Gliwice-Cracow, 1993
19. T. Łuba: *Multi-level logic synthesis based on decomposition.* Microprocessors and Microsystems, Vol. 18, No. 8, Octomber 1994, pp. 429–437
20. G. de Micheli: *Synthesis and optimization of digital circuits.* USA, McGraw-Hill, 1994
21. P. Michel, U. Lauther, P. Duzy: *The Synthesis Approach to Digital System Design.* Boston, Kluwer Academic Publishers, 1993
22. J.P. Roth, R.M. Karp: *Minimization over Boolen graphs.* IBM Journal of Research and Development, Vol.6, April 1962, pp. 227–238
23. G. Saucier, P. Sicard, L. Bouchet: *Multi-level synthesis on PAL's.* Proc. European Design Automation Conference, Glasgow, March 1990, pp. 542–546
24. T. Sasao: *FPGA Design by Generalized Functional Decomposition in Logic Synthesis and Optimization.* Boston, Kluwer Academic Publishers, 1993
25. H. Selveraj, T. Łuba, M. Nowicka, B. Bignall: *Multiple-valued decomposition and its applications in data compression and technology mapping.* ICCIMA'97
26. F. Volf: *A bottom-up approach to multiple-level synthesis for look-up table based FPGAs.* CIP-Data Library Technische Iniversiteit Eindhoven 1997
27. W. Wan, M.A. Perkowski: *A new Approach to the Decomposition of Incompletely Specified Multi-Output Functions Based on Graph Coloring and Local Transformations and Its Applications to FPGA Mapping.* Proceedings EDAC'92, Bruesels

D. KANIA

LOGIC SYNTHESIS ON PAL-BASED DEVICES CONTAINING OUTPUT BUFFERS

Summary

The PAL-based structure constitutes the kernel of many CPLDs. The problem of appropriate partition of the whole devices under design into suitable parts, which can be implemented as single PAL-based logic blocks containing the limited number of terms, is one of basic problems of the synthesis process. The method of logic synthesis that makes use of three-state output buffers constituting the additional internal resources of logic blocks, is presented in this paper. Developed algorithms have been implemented and used for partitioning the benchmark circuits due to realization by means of the PAL-based logic blocks with the given number of terms. Synthesis of benchmark circuits for PAL devices has also been carried out and the obtained results have been compared to classical approach and the ones published in bibliography.

Keywords: logic synthesis, partitioning, decomposition, minimization, PLD

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2002, 48, z. 1, ss. 67–80

Synteza paskowych szyków antenowych na podłożu ferroelektrycznym

EUGENIUSZ JASZCZYSZYN, JÓZEF MODELSKI

Instytut Radioelektroniki, Politechnika Warszawska
Nowowiejska 15/19, 00-665, Warszawa
e-mail: e.jaszczyszyn@ire.pw.edu.pl; j.modelski@ire.pw.edu.pl

Otrzymano 2001.07.20
Autoryzowano 2001.09.20

W pracy przedstawiono nową metodę syntezy anten paskowych, przydatnych do zastosowań w systemach adaptacyjnych. Proponowana metoda polega na poszukiwaniu optymalnych kształtów charakterystyk promieniowania pojedynczych radiatorów w szyku antenowym (przez zmianę przenikalności elektrycznej podłoża ferroelektrycznego radiatora paskowego) bez konieczności stosowania przesuwników fazy i tłumików. W pracy przedstawione wyniki syntezy oraz porównanie proponowanej metody z ogólnie stosowanymi.

Słowa kluczowe: anteny paskowe, szyki antenowe, ferroelektryki

1. WPROWADZENIE

Gwałtowny rozwój systemów radiokomunikacyjnych narzuca potrzebę pracy w coraz wyższych pasmach częstotliwości oraz pokonywanie kolejnych barier technologicznych, ponieważ nowoczesne systemy radiokomunikacyjne charakteryzują się coraz większą złożonością funkcjonalną i układową. W takich systemach konieczne jest stosowanie anten inteligentnych (ang. *smart antenna*). W ostatniej dekadzie stanowią one najbardziej dynamicznie rozwijającą się dziedzinę techniki antenowej. Anteny inteligentne mogą dostosowywać się do zewnętrznych warunków w przestrzeni elektromagnetycznej, na przykład, do dynamicznie powstających zakłóceń. Problem kształtowania charakterystyki promieniowania można rozwiązywać, m.in. poprzez stosowanie fazowanych szyków antenowych, które mogą sterować kierunkiem nadawania i odbioru sygnałów bez konieczności mechanicznej zmiany położenia systemu antenowego w przestrzeni. Często takie szyki składają się z dużej ilości radiatorów, które są zasilane

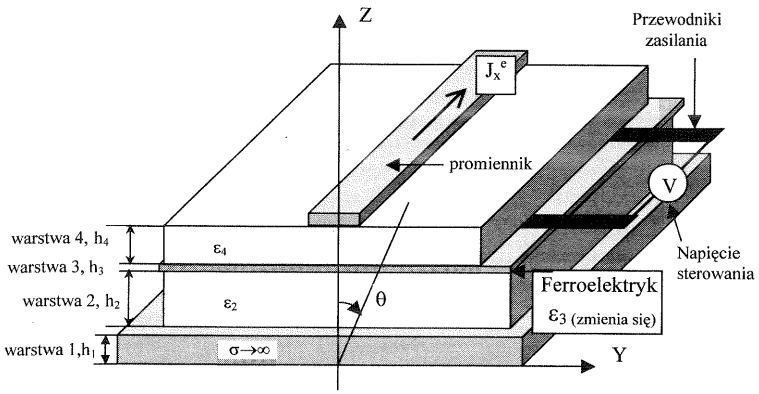
ze zmienną fazą lub z opóźnieniem czasowym. Ukształtowanie odpowiedniego rozkładu amplitudowo-fazowego na aperturze szyku pozwala na uformowanie wiązki o pożądanym kształcie. Możliwości fazowanych szyków antenowych są dobrze znane, a ich główną wadę stanowią znaczne koszty podzespołów (przesuwników fazy, tłumików, dzielników mocy, itd.). Oprócz tego każdy radiator szyku antenowego z osobna może silnie wpływać na proces kształtowania wiązki. Ze względu na rozmiary, objętość, koszty systemu antenowego oraz możliwości łatwego sterowania charakterystykami anteny, konieczne jest opracowanie i badanie nowych, tanich i łatwo sterowanych elementów promieniujących oraz szyków antenowych. Z drugiej jednak strony, obecnie stosowane metody adaptacji w systemach antenowych wykorzystują uproszczone traktowanie szyków antenowych, a w szczególności nie uwzględniają związków wzajemnych między radiatorami. Z reguły nie są też uwzględnione kierunkowe i polaryzacyjne charakterystyki poszczególnych elementów anten inteligentnych. Celem niniejszej pracy jest przedstawienie metody syntezy anten paskowych, przydatnych do zastosowań w systemach adaptacyjnych. Proponowana metoda syntezy polega na poszukiwaniu optymalnych kształtów charakterystyk promieniowania pojedynczych radiatorów w szyku antenowym bez konieczności stosowania przesuwników fazy i tłumików. Realizacja tej metody jest możliwa dzięki zastosowaniu sterowanych radiatorów paskowych na podłożu z dielektrykiem nieliniowym [L-1, 2]. Ze względu na konwencjonalny kształt anteny paskowej, elementem sterującym może być jej podłoże, wykonane na przykład z materiału aktywnego. Materiał ten powinien zawierać cząstki posiadające możliwości „magazynowania" energii. Takie właściwości mają nieliniowe materiały elektrotechniczne – ferromagnetyki i ferroelektryki. Porównanie właściwości dwóch typów materiałów nieliniowych – ferroelektryków i ferromagnetyków, ze względu na ich wykorzystanie w procesie sterowania, pozwala wyciągnąć wnioski na korzyść ferroelektryków. W szczególności moc sterowania ferroelektryków jest o rząd wielkości (lub nawet dwa) niższa niż w przypadku ferromagnetyków.

2. KONSTRUKCJA ANTENY

Idea sterowanej anteny ferroelektrycznej polega na możliwości zmiany skutecznej przenikalności elektrycznej podłoża anteny paskowej, poprzez zmianę zewnętrznego sterującego pola elektrycznego E_0. Podłoże takiej anteny może być wykonane w postaci kompozytowej struktury zawierającej dielektryk liniowy z lokalnym wtrąceniem ferroelektryka lub w postaci podłoża wielowarstwowego z cienką warstwą ferroelektryka (rys. 1).

Wielowarstwowa struktura pokazana na rys. 1 może składać się z dwóch warstw typowego dielektryka (warstwy 2 i 4), między którymi znajduje się cienka warstwa ferroelektryka (warstwa 3). Dielektryki są umieszczone na metalowym ekranie (tak jak w konwencjonalnych antenach paskowych).

Zmiana skutecznej przenikalności elektrycznej podłoża odbywa się poprzez zmianę przenikalności elektrycznej warstwy ferroelektryka. Taka zmiana jest możliwa, ponie-



Rys. 1. Konstrukcja anteny ferroelektrycznej

waż przy pomocy zewnętrznego pola elektrycznego można polaryzować ferroelektryk. W rzeczywistości dla zabezpieczenia pożądanej zmiany przenikalności potrzebna jest zmiana napięcia sterowania do kilku kilowoltów. Trzeba jednak zauważyć, ze prąd sterowania jest nieznaczny, ponieważ mamy do czynienia z dobrym dielektrykiem. Dlatego moc sterowania nie będzie przekraczać poziomu rzędu 5mW [L-3]. W ogólnym przypadku interesuje nas zmiana przenikalności elektrycznej podłoża w kierunku osi OZ, dlatego dalsza analiza może być przeprowadzona przy założeniu izotropowej przenikalności elektrycznej ferroelektryka (w rzeczywistości przenikalność elektryczna spolaryzowanego ferroelektryka jest anizotropowa).

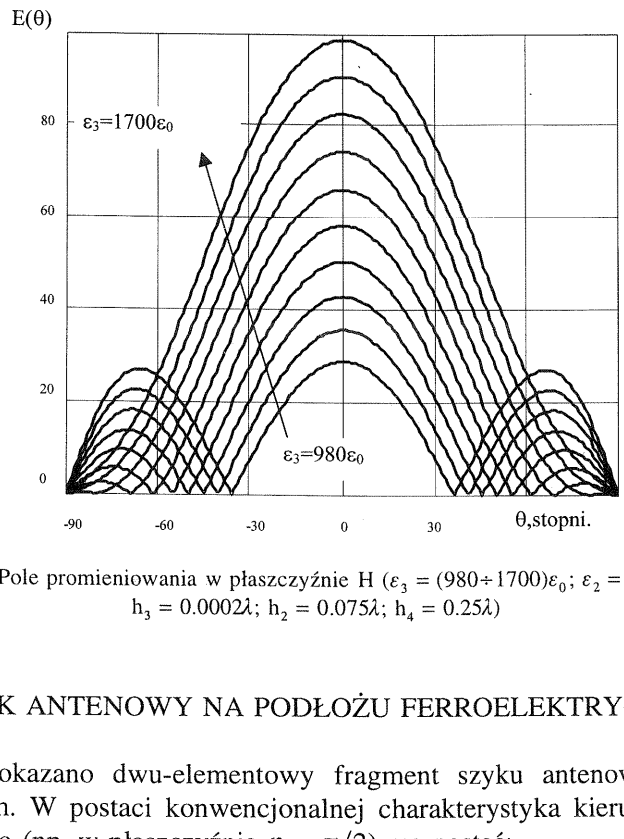
Jako promiennik może być stosowany przewodnik o dowolnym kształcie. W celu pokazania ogólnych możliwości sterowanych anten paskowych kształt promiennika nie jest na tym etapie ważny i można przyjąć najprostszy przypadek.

3. ANALIZA ANTENY

Do analizy struktury wielowarstwowej pokazanej na rys. 1 wykorzystano metodę pełnofalową, bazującą na obliczaniu pola elektromagnetycznego wzbudzanego dowolnym źródłem wewnątrz ośrodka wielowarstwowego.

Charakterystyka promieniowania jest podstawową cechą każdej anteny ze względu na możliwość zmiany jej kształtu. Struktura proponowanej anteny daje możliwość wyboru sześciu parametrów podłoża (h_2, h_3, h_4, ε_2, ε_3, ε_4). Załóżmy, że warstwa ferroelektryka jest cienka i ma dużą wartość przenikalności elektrycznej (rzędu $1000\varepsilon_0$ [L-3, 4]). Przenikalności warstw drugiej i czwartej powinny być analogiczne do przenikalności podłoża anteny konwencjonalnej (np., $\varepsilon_2 = \varepsilon_4 = 2\varepsilon_0$). W niniejszej pracy rozpatrujemy ośrodek niemagnetyczny a pole w strefie dalekiej zostało obliczone stosując metodę fazy stacjonarnej. Pole promieniowania w płaszczyźnie H rozpatrywanej anteny (dla różnych wartości przenikalności elektrycznej warstwy ferroelekt-

ryka) pokazano na rys. 2. Z rysunku tego widać, że można znacznie zmieniać kształt charakterystyk promieniowania, co jest bardzo ważną cechą ze względu na możliwość budowania sterowanych anten. Przedstawione zależności otrzymane dla takich parametrów podłoża: warstwa 1 (przewodnik) – $h_1 = 0.02\lambda$ oraz $\sigma \to \infty$; warstwa 2 – $h_2 = 0.075\lambda$ oraz $\varepsilon_2 = 2 \cdot \varepsilon_0$; warstwa 3 (ferroelektryk) – $h_3 = 0.0002\lambda$ oraz $\varepsilon_{3i} = (980 + 72 \cdot u) \cdot \varepsilon_0$, gdzie „u" może zmieniać się od 0 do 10; warstwa 4 – $h_4 = 0.25\lambda$ oraz $\varepsilon_4 = 2 \cdot \varepsilon_0$, gdzie λ – długość fali. Oprócz zmiany kształtu wiązki udaje się też uzyskać możliwość sterowania punktem zerowym charakterystyki promieniowania pojedynczej anteny [L-5].



Rys. 2. Pole promieniowania w płaszczyźnie H ($\varepsilon_3 = (980 \div 1700)\varepsilon_0$; $\varepsilon_2 = \varepsilon_4 = 2\varepsilon_0$; $h_3 = 0.0002\lambda$; $h_2 = 0.075\lambda$; $h_4 = 0.25\lambda$)

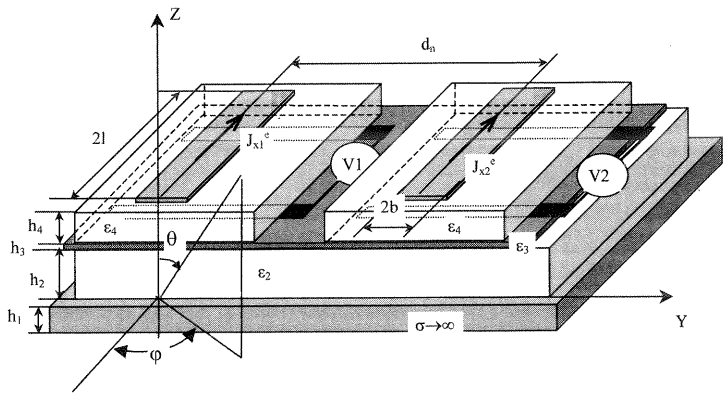
4. SZYK ANTENOWY NA PODŁOŻU FERROELEKTRYCZNYM

Na rys. 3 pokazano dwu-elementowy fragment szyku antenowego na podłożu ferroelektrycznym. W postaci konwencjonalnej charakterystyka kierunkowa liniowego szyku antenowego (np. w płaszczyźnie $\varphi = \pi/2$) ma postać:

$$F(\theta) = \sum_{n=1}^{N} f_n(\theta) e^{ik_0 d_n \sin\theta}, \tag{1}$$

gdzie $f_n(\theta)$ – charakterystyka kierunkowa pojedynczego promiennika; $k_0 = 2\pi/\lambda$.

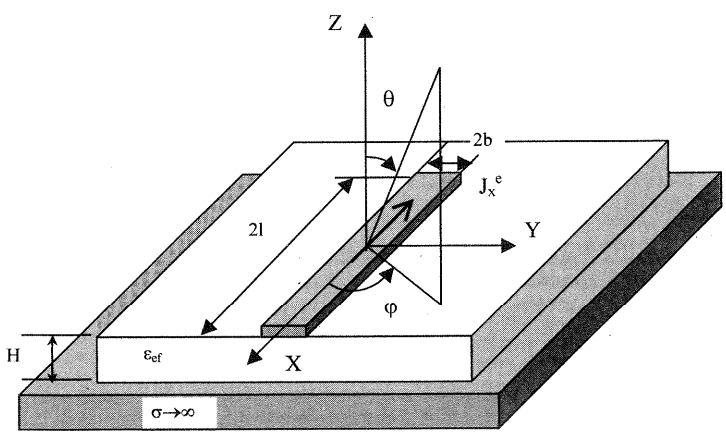
Jeśli kształt charakterystyki kierunkowej pojedynczego promiennika w szyku może być zmieniony (przez zmianę jakiegokolwiek parametru struktury, w naszym przypadku przez zmianę przenikalności podłoża) wtedy istnieje możliwość syntezy charakterystyki



Rys. 3. Dwuelementowy fragment szyku antenowego

kierunkowej szyku antenowego bez konieczności stosowania przesuwników fazy i tłumików dla uzyskania pożądanego pobudzania radiatorów. Na tym polega proponowana metoda syntezy, czyli na poszukiwaniu odpowiednich kształtów charakterystyk promieniowania pojedynczych promienników $f_n(\theta)$, które w układzie antenowym dadzą pożądany kształt charakterystyki promieniowania szyku antenowego. Analizę pola promieniowania paskowego szyku antenowego w zależności od przenikalności elektrycznej warstwy ferroelektryka można przeprowadzić wykorzystując wyniki, otrzymane w [L-6]. Analiza struktury wielowarstwowej może być także przeprowadzona przez analizę jednowarstwowego podłoża zastępczego przenikalność elektryczna którego jest równa skutecznej przenikalności elektrycznej podłoża wielowarstwowego ε_{ef}.

Załóżmy, że wzdłuż promiennika istnieje sinusowy rozkład prądu a poprzek – równomierny (rys. 4).



Rys. 4. Struktura zastępcza

W wyniku rozwiązywania przedstawionego zagadnienia brzegowego elektrodynamiki można otrzymać wszystkie składowe pola. Elektryczne składowe mają postać

$$E_i = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} E_{i\chi}\mathrm{d}\chi_1\mathrm{d}\chi_2, \tag{2}$$

gdzie:

$$E_{x\chi} = i\frac{W}{4\pi^2 k_0}\left[\frac{\chi_1^2\gamma_1\gamma_2}{\varepsilon_{ef}\gamma_1 cth(H\gamma_2)+\gamma_2} - \frac{k_0^2\chi_2^2}{\gamma_1+\gamma_2 cth(H\gamma_2)}\right]\frac{e^{-i\chi_1 x - i\chi_2 y - \gamma_1 z}}{\chi_1^2+\chi_2^2} I(\nu');$$

$$E_{y\chi} = i\frac{W}{4\pi^2 k_0}\left[\frac{\gamma_1\gamma_2}{\varepsilon_{ef}\gamma_1 cth(H\gamma_2)+\gamma_2} + \frac{k_0^2}{\gamma_1+\gamma_2 cth(H\gamma_2)}\right]\frac{\chi_1\chi_2 e^{-i\chi_1 x - i\chi_2 y - \gamma_1 z}}{\chi_1^2+\chi_2^2} I(\nu');$$

$$E_{z\chi} = \frac{W}{4\pi^2 k_0}\left[\frac{\chi_1\gamma_2}{\varepsilon_{ef}\gamma_1 cth(H\gamma_2)+\gamma_2}\right] e^{-i\chi_1 x - i\chi_2 y - \gamma_1 z} I(\nu');$$

$$\gamma_1 = \sqrt{\chi_1^2+\chi_2^2-k_0^2}\,; \quad \gamma_2 = \sqrt{\chi_1^2+\chi_2^2-\varepsilon_{ef}k_0^2}$$

$$I(\nu') = I_x^e 2k_0\left[\frac{\sin(\chi_2 b)}{\chi_2 b}\right]\left[\frac{\cos(\chi_1 l)-\cos(k_0 l)}{(k_0^2-\chi_1^2)\sin(k_0 l)}\right].$$

Z kolei moc promieniowania można przedstawić w postaci

$$P(\theta,\varphi) = \frac{60\cdot(I_x^e)^2}{\pi\cdot\sin^2(2\pi l)}\cdot\left\{\frac{(\varepsilon_{ef}-\sin^2\theta)\cdot\cos^2\varphi}{T_m(\theta)} + \frac{\sin^2\varphi}{T_e(\theta)}\right\}\cdot\cos^2\theta\cdot\Psi^2(b,l,\theta,\varphi), \tag{3}$$

gdzie:

$$\Psi^2(b,l,\theta,\varphi) = \left\{\frac{\sin(2\pi b\cdot\sin\theta\sin\varphi)}{2\pi b\cdot\sin\theta\sin\theta}\right\}^2\cdot\left\{\frac{\cos(2\pi l\cdot\sin\theta\cos\varphi)-\cos(2\pi l)}{1-\sin^2\theta\cos^2\varphi}\right\}^2;$$

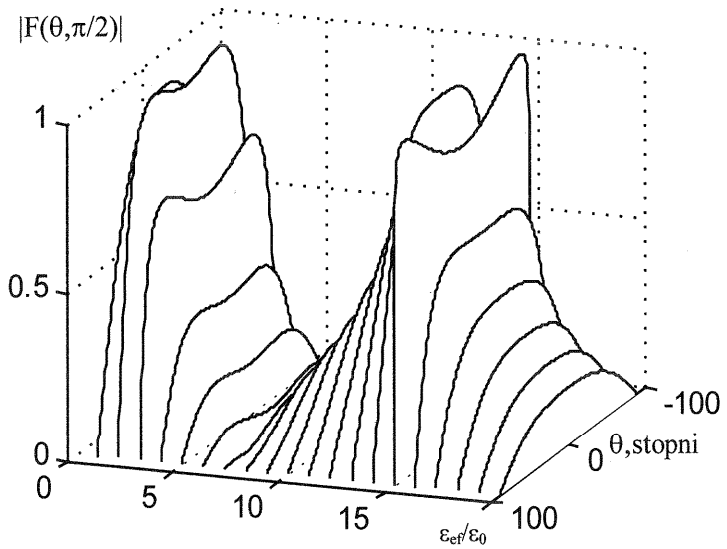
$$T_m(\theta) = \varepsilon_{ef}^2\cos^2\theta\cdot\mathrm{ctg}^2\left(2\pi H\sqrt{\varepsilon_{ef}-\sin^2\theta}\right)+(\varepsilon_{ef}-\sin^2\theta);$$

$$T_e(\theta) = (\varepsilon_{ef}-\sin^2\theta)\cdot ctg^2\left(2\pi H\sqrt{\varepsilon_{ef}-\sin^2\theta}\right)+(\cos^2\theta);$$

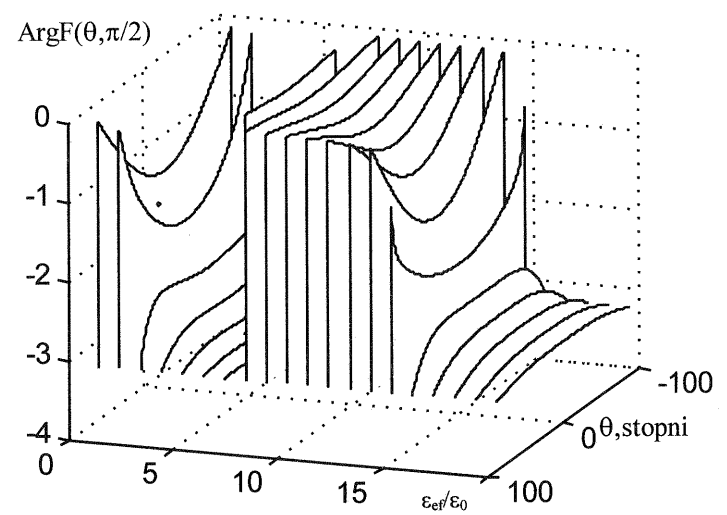
$$H = h_2+h_3+h_4.$$

Na rys. 5 i rys. 6 pokazane wyliczone amplitudowe oraz fazowe charakterystyki promieniowania w płaszczyźnie ZOY pojedynczego promiennika w zależności od przenikalności elektrycznej podłoża ε_{ef}. Z rysunków widać, że amplituda promieniowania zmienia się od 0 do 1 a faza od 0 do $-\pi$ w kierunku pionowym do apertury anteny.

Takie zachowanie charakterystyki promieniowania jest bardzo ważne dla stosowania proponowanej metody syntezy szyku antenowego.



Rys. 5. Amplitudowe charakterystyki promieniowania w zależności od przenikalności elektrycznej podłoża



Rys. 6. Fazowe charakterystyki promieniowania w zależności od przenikalności elektrycznej podłoża

Rozwiązanie zagadnienia syntezy szyku antenowego jest możliwe stosując techniki numeryczne przez optymalizacje wybranej funkcji celi. Optymalizacja w naszym przypadku polega na minimalizacji funkcjonału $f(\varepsilon_n)$ przez poszukiwanie wektora przenikal-

ności elektrycznych $\{\varepsilon_n\}$ podłoża poszczególnych promienników. Procedura iteracyjna w postaci

$$f(\varepsilon_{n+1}) < f(\varepsilon_n), \tag{4}$$

pozwala na odnalezienie minimum lokalnego funkcjonału. Rozpatrzmy dwa zagadnienia syntezy. Pierwsze polega na syntezie szyku antenowego, który powinien wzbudzać pole elektromagnetyczne o pożądanym kształcie charakterystyki kierunkowej. Drugie zaś na stworzeniu zera w charakterystyce kierunkowej w pożądanym kierunku. Funkcjonały dla rozpatrywanych zagadnień można zapisać w postaci:

1) Dla kształtowania pożądanej wiązki:

$$f(\varepsilon_n) = \|\Phi(\theta) - F(\theta)\|^2, \tag{5}$$

gdzie $\Phi(\theta)$ – pożądany kształt wiązki;

2) Dla stworzenia zera (lub dużego tłumienia) w charakterystyce kierunkowej w pożądanym kierunku:

$$f(\varepsilon_n) = \frac{F^2(\theta_i)}{F^2(\theta_0)} + \alpha \cdot \frac{1}{D}, \tag{6}$$

gdzie θ_i – kierunek tłumienia; θ_0 – kierunek pożądanego sygnału; D – kierunkowość; α – współczynnik priorytetu.

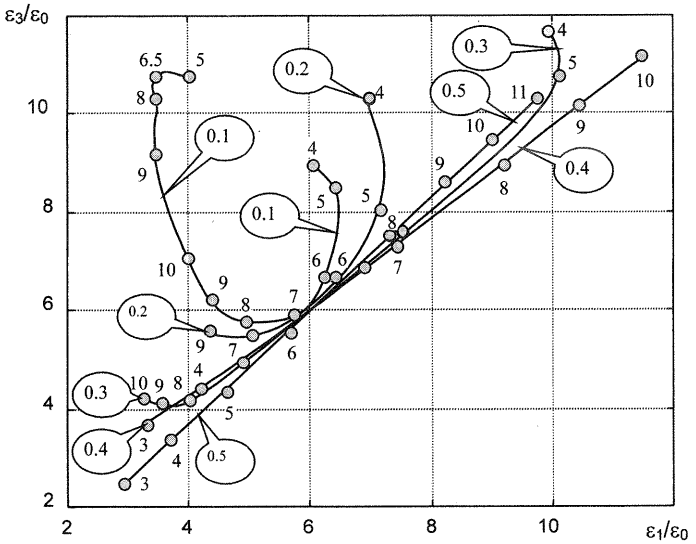
Załóżmy, że szyk antenowy składa się z radiatorów na podłożu ferroelektrycznym a charakterystyki kierunkowe poszczególnych radiatorów nie są korelowane oraz mogą mieć różny kształt. Charakterystyka kierunkowa takiego szyku może być zapisana w postaci

$$F(\theta, \varphi) = \sum_{n=1}^{N} I_n |f_n(\theta, \varphi)| \cdot \exp\left[i\psi_n(\theta,\varphi)\right] \cdot \exp\left[ik_0 d_n \sin\theta\right], \tag{7}$$

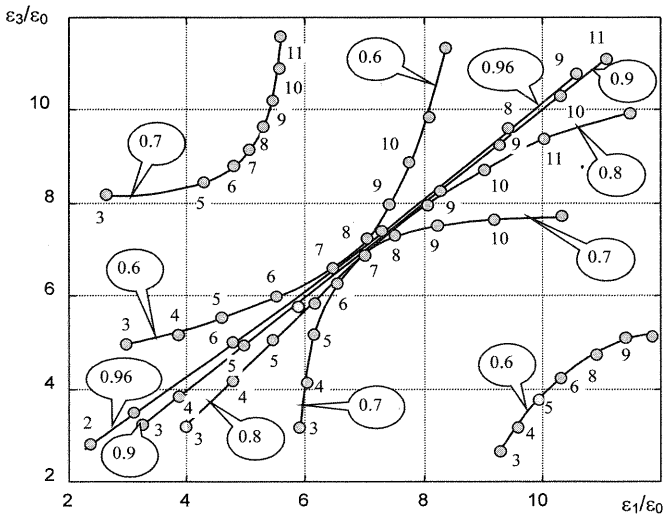
gdzie d – odległość między radiatorami; I_n – amplitudy zespolone pobudzania oraz $|f_n(\theta,\varphi)| \cdot \exp[i\psi_n(\theta,\varphi]$ – charakterystyka kierunkowa n-go radiatora. Jeśli wszystkie promienniki są jednakowe (w naszym przypadku są jednakowe przenikalności elektryczne podłoża poszczególnych radiatorów) wtedy charakterystyka promieniowania może być obliczona ze wzoru

$$F(\theta, \varphi) = |f_0(\theta, \varphi)| \cdot \exp\left[i\psi_0(\theta,\varphi)\right] \cdot \sum_{n=1}^{N} I_n \exp\left[ik_0 d_n \sin\theta\right], \tag{8}$$

gdzie $\sum_{n=1}^{N} I_n \exp[ik_0 d_n \sin\theta]$ – współczynnik szyku lub charakterystyka promieniowania



Rys. 7. Lokalizacja zera w charakterystyce promieniowania ($\sin\theta_i = 0.1 \div 0.5$) trzyelementowego szyku antenowego dla różnych wartości przenikalności elektrycznej podłoża poszczególnych radiatorów



Rys. 8. Lokalizacja zera w charakterystyce promieniowania ($\sin\theta_i = 0.6 \div 0.96$) trzyelementowego szyku antenowego dla różnych wartości przenikalności elektrycznej podłoża poszczególnych radiatorów

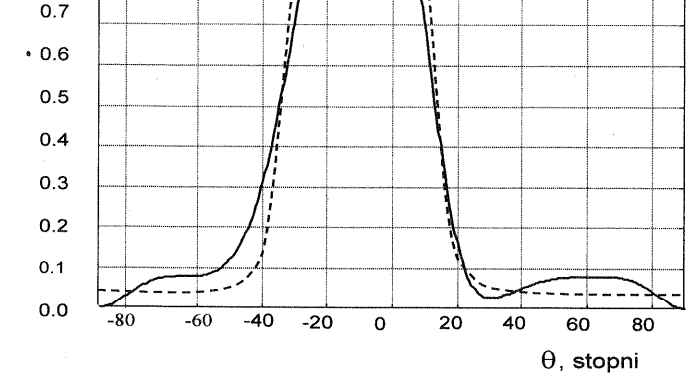
izotropowych promienników. Wiadomo, że liniowy szyk N-izotropowych radiatorów może mieć (N-1) stopni wolności co pozwala stwarzać (N-1) kontrolowanych zer w charakterystyce promieniowania [L-7, 8]. Jeśli charakterystyka promieniowania nie-izotropowego radiatora ma własne kontrolowane zera, wtedy charakterystyka promienio-

wania szyku może mieć (N+1) kontrolowanych zer. Taka cecha może być wykorzystana dla kształtowania zer w charakterystyce promieniowania w pożądanych kierunkach.

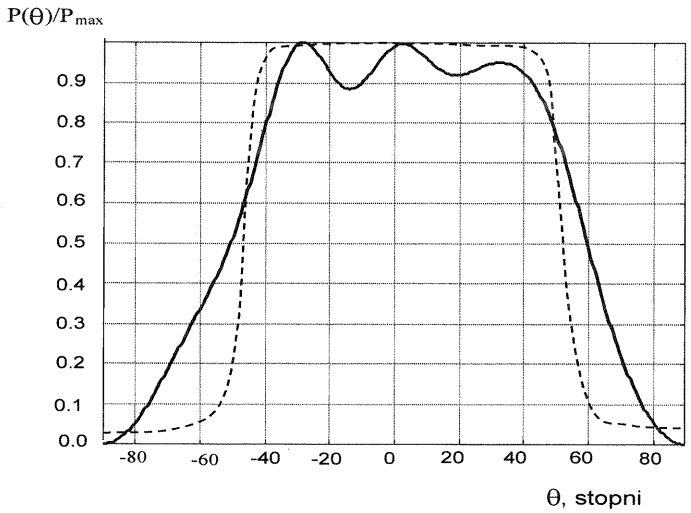
Na rys. 7 i rys. 8 przedstawiona możliwa lokalizacja zer w charakterystyce promieniowania trzy-elementowego szyku antenowego na podłożu ferroelektrycznym ($d = 0.75\lambda$) dla różnych wartości przenikalności elektrycznej (zastępczej) podłoża poszczególnych radiatorów. Przenikalność ε_1 zmienia się wzdłuż osi poziomej, ε_3 – wzdłuż osi pionowej oraz ε_2 przedstawiona na rysunku liczbami 4, 5, ... 10 .... Wykresy na rysunkach są podane dla różnych wartości $\sin\theta_i$, gdzie θ_i jest kierunek pożądanej lokalizacji zera w charakterystyce promieniowania. Wartości $\sin\theta_i$ zmieniają się od 0.1 do 0.5 na rys.7 oraz od 0.6 do 0.96 na rys.8.

5. WYNIKI SYMULACJI I PORÓWNANIA

Na rys. 9 przedstawiono przykład pożądanej (linia przerywana) i po syntezie (linia ciągła) charakterystyki promieniowania otrzymanej od sześcioelementowego szyku antenowego ($d/\lambda = 0.5$; $h/\lambda = 0.2$; przenikalność poszczególnych podłoży wynosi: $\varepsilon_1/\varepsilon_0 = 15.65$; $\varepsilon_2/\varepsilon_0 = 14.06$; $\varepsilon_3/\varepsilon_0 = 13.43$; $\varepsilon_4/\varepsilon_0 = 22.0$; $\varepsilon_5/\varepsilon_0 = 5.18$; $\varepsilon_6/\varepsilon_0 = 22.0$). Założenia do syntezy były: wiązka pożądana z płaskim wierzchołkiem o szerokości $2\Delta\theta_{0.5} \approx 45^0$ z maksimum wiązki w kierunku -10^0. Podobne wyniki syntezy przedstawione na rys. 10 dla wiązki o szerokości $2\Delta\theta_{0.5} \approx 95^0$ z maksimum w kierunku $+5^0$. Przenikalność poszczególnych podłoży dla tego przypadku wynosi: $\varepsilon_1/\varepsilon_0 = 5.49$; $\varepsilon_2/\varepsilon_0 = 22.0$; $\varepsilon_3/\varepsilon_0 = 13.75$; $\varepsilon_4/\varepsilon_0 = 5.49$; $\varepsilon_5/\varepsilon_0 = 22.0$; $\varepsilon_6/\varepsilon_0 = 6.13$.

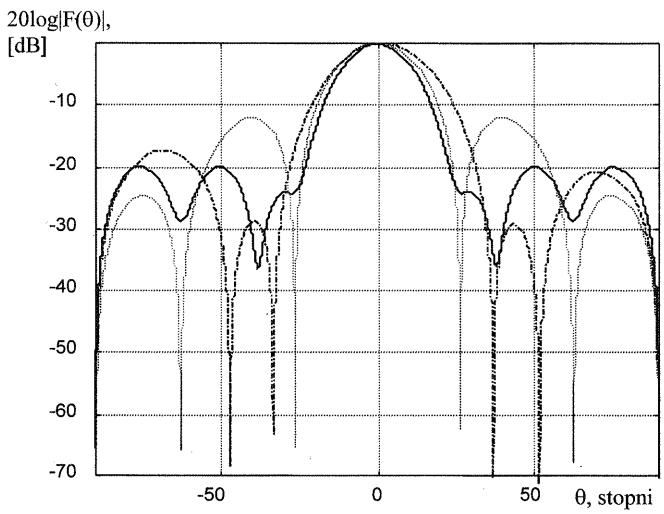


Rys. 9. Pożądana (linia przerywana) i syntezowana (linia ciągła) charakterystyki promieniowania sześcioelementowego szyku ($d/\lambda = 0.5$; $h/\lambda = 0.2$; Przenikalność elektryczna poszczególnych podłoży: $\varepsilon_1/\varepsilon_0 = 15.65$; $\varepsilon_2/\varepsilon_0 = 14.06$; $\varepsilon_3/\varepsilon_0 = 13.43$; $\varepsilon_4/\varepsilon_0 = 22.0$; $\varepsilon_5/\varepsilon_0 = 5.18$; $\varepsilon_6/\varepsilon_0 = 22.0$)



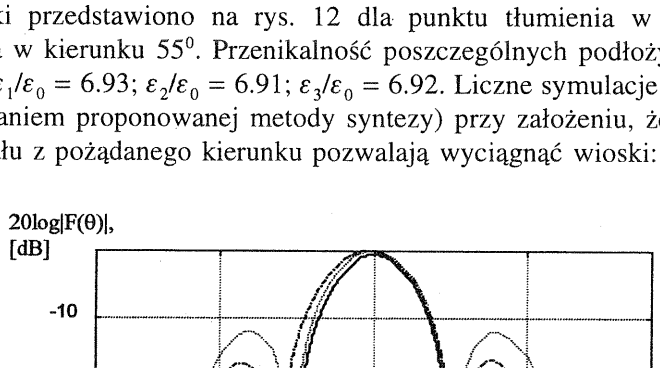
Rys. 10. Pożądana (linia przerywana) i syntetyzowana (linia ciągła) charakterystyki promieniowania sześcioelementowego szyku (d/λ = 0.5; h/λ = 0.2; Przenikalność elektryczna poszczególnych podłoży: $\varepsilon_1/\varepsilon_0 = 5.49$; $\varepsilon_2/\varepsilon_0 = 22.0$; $\varepsilon_3/\varepsilon_0 = 13.75$; $\varepsilon_4/\varepsilon_0 = 5.49$; $\varepsilon_5/\varepsilon_0 = 22.0$; $\varepsilon_6/\varepsilon_0 = 6.13$)

Na rys. 11 przedstawiono przykład charakterystyki kierunkowej trzyelementowego szyku ($\varepsilon_1/\varepsilon_0 = 6.66$; $\varepsilon_2/\varepsilon_0 = 6.61$; $\varepsilon_3/\varepsilon_0 = 6.57$) z założeniem do syntezy – stworzenie punktu tłumienia w charakterystyce promieniowania w kierunku 37^0. Oprócz tego na rys. 11 przedstawiono charakterystyki promieniowania trzyelementowego synfazowa-



Rys. 11. Charakterystyki promieniowania trzyelementowego szyku ferroelektrycznego z tłumieniem sygnału w kierunku 37^0. Przenikalności elektryczne: $\varepsilon_1/\varepsilon_0 = 6.66$; $\varepsilon_2/\varepsilon_0 = 6.61$; $\varepsilon_3/\varepsilon_0 = 6.57$ (linia ciągła – dla szyku syntetyzowanego, przerywana – dla szyku Applebauma'a, kropkowana – dla szyku współfazowanego)

nego szyku antenowego do i po adaptacji metodą Applebaum'a [L-9, 10]. Charakterystyka promieniowania pojedynczego radiatora w konwencjonalnym szyku antenowym była przyjęta w postaci $f_0(\theta) = cos(\theta)$ a odległość między radiatorami wynosiła 0.75λ. Podobne wyniki przedstawiono na rys. 12 dla punktu tłumienia w charakterystyce promieniowania w kierunku 55^0. Przenikalność poszczególnych podłoży dla tego przypadku wynosi: $\varepsilon_1/\varepsilon_0 = 6.93$; $\varepsilon_2/\varepsilon_0 = 6.91$; $\varepsilon_3/\varepsilon_0 = 6.92$. Liczne symulacje (przeprowadzone z wykorzystaniem proponowanej metody syntezy) przy założeniu, że konieczne jest tłumienie sygnału z pożądanego kierunku pozwalają wyciągnąć wioski:



Rys. 12. Charakterystyki promieniowania trzyelementowego szyku ferroelektrycznego z tłumieniem sygnału w kierunku 55^0. Przenikalności elektryczne: $\varepsilon_1/\varepsilon_0 = 6.93$; $\varepsilon_2/\varepsilon_0 = 6.91$; $\varepsilon_3/\varepsilon_0 = 6.92$ (linia ciągła – dla szyku syntetyzowanego, przerywana – dla szyku Applebauma'a, kropkowana – dla szyku współfazowanego)

- Kierunkowość D_s wiązki syntetyzowanej jest wyższa od kierunkowości charakterystyki promieniowania otrzymanej po zastosowaniu metody Applebaum'a D_a, np. dla charakterystyk pokazanych na rys. 11 $D_s = 10.3191$ i $D_a = 9.2771$ oraz $D_s = 10.1327$dB i $D_a = 9.5561$dB. na rys.12.
- Szerokość wiązki syntetyzowanej jest mniejsza od szerokości charakterystyki promieniowania otrzymanej po stosowaniu metody Applebaum'a i prawie jest równa szerokości wiązki szyku współfazowanego.
- Poziom listków bocznych wiązki syntetyzowanej jest mniejszy od poziomu listków bocznych charakterystyki promieniowania otrzymanej po stosowaniu metody Applebaum'a.
- Tłumienie sygnału w założonych kierunkach, otrzymane przez zastosowanie proponowanej metody jest mniejsze w porównaniu do tłumienia uzyskanego po stosowaniu metody Applebaum'a.

6. PODSUMOWANIE

W pracy została przedstawiona metoda syntezy paskowych szyków antenowych na podłożu ferroelektrycznym bez konieczności stosowania przesuwników fazy i tłumików. Proponowana metoda polega na poszukiwaniu optymalnych (według założonych kryteriów) kształtów charakterystyk promieniowania pojedynczych radiatorów w szyku antenowym przez zmianę zastępczej przenikalności elektrycznej podłoża poszczególnych radiatorów. Opisana w pracy metoda syntezy może znaleźć zastosowanie w stacjach bazowych współczesnych systemów radiokomunikacyjnych.

7. BIBLIOGRAFIA

1. J. Modelski, Y. Yashchyshyn: *Voltage-controlled ferroelectric microstrip antenna for phased arrays*. 2000 IEEE Intern. Symp. on Antennas and Propagation and USNC/URSI National Radio Science Meeting., Utah, Salt Lake City (USA), July 2000, pp. 506-509
2. J. Modelski, Y. Yashchyshyn: *Investigations of the Microstrip Antenna on Ferroelectric Substrates*. 30th European Microwave Conference, Paris (France), October 2000, pp. 162-165
3. J. B. L. Rao, D. P. Patel and V. Krichevsky: *Voltage-Controlled Ferroelectric Lens Phased Arrays*. IEEE Trans. Antennas and Propagation, 1999, AP-47, No 3, pp. 458-468
4. V.K. Varadan, D. K. Ghodgaonkar, V.V. Varadan, J.F. Kelly and P. Glikerdas: *Ceramic Phase Shifters for Electronically Steerable Antenna Systems*. Microwave Journal, 1992, MJ-35, No 1, pp. 116-127
5. J. Modelski, Y. Yashchyshyn: *New type of Microstrip Antenna with Ferroelectric Layer*. XIII Intern. Conf. On Microwaves, Radar and Wireless Communications MIKON'2000, Wroslaw (Poland), May 2000, pp. 583-586
6. Є. М. Ящишин: *Сегнетоелектрична мікросмужкова антена*. Вісник НУ"ЛП" №387 „Радіоелектроніка, та телекомунікації", Львів (Україна), 2000, сс. 222-231
7. R. C. Hansen: *Phased Array Antennas*. John Wiley & Sonc, Inc. 1998, p. 486
8. R. J. Mailloux: *Phased Array Antenna Handbook*. Artech House. 1994, p. 525
9. Р. А. Монзиго, Т. У. Миллер: *Агаптивные антенные решетки*. Введеные в теорию. Пер. с англ. под ред. В.А. Лексаченко. М. Радио и связь, 1986, с. 446
10. Б. Уидроу, С. Стирнз: Адаптивная обработка сугналов. Пер. с англ. Ю.К. Сальникова под ред. В.В. Шахгильдяна. М. Радио и связь, 1989, с. 440

Y. YASHCHYSHYN J. MODELSKI

SYNTHESIS OF MICROSTRIP ANTENNA ARRAYS ON FERROELECTRIC SUBSTRATE

Summary

In modern radio system smart beamforming is often required. Communications antennas must offer a large number of operating modes (including pencil and shaped beams with fast switching between them) in order to ensure the best coverage of the service area. Therefore in designing smart antennas one of the main challenges is to provide a prescribed shaped antenna pattern that simultaneously suppresses interference

signals (which locations are either known or unknown). As a rule this problem is solved with the use of phased-array antennas. This type of array consists of multiple stationary antenna elements which are fed coherently and use variable phase or time-delay control at each element to scan the beam to given angles in space. Variable amplitude control is sometimes also provided for pattern shaping. Perfrmance of each radiating element has significant influence on the parameters of antenna array, e.g. pattern, beamwidth, directivity gain, sidelobes, cross-polarization levels etc.

Reducing the cost of phased array antennas has recently become the subject of considerable interest. The most prominent obstacle to this is the cost of current phase shifter elements (p-I-n diodes, MESFETs, ferrite). All of these are expensive, constituting a significant portion of the total receive-array cost. Additionally, p-I-n diodes and MESFETs introduce significant losses. These losses necessitate additional amplification of signal thus complicate the overall system design. New developments in MEMS and ferroelectric technologies promise low loss/high performance phase shifters. Both technologies provide low losses with beam capabilities, high isolation, good power handling ability, and low intermodulation distortion. Ferroelectrics are beneficial for applications operating at microwaves range, offering large phase shift per loss, low losses (0.2–0.3 dB, depending on frequency and matching), and large power handling capability. Bulk ferroelectric are beneficial for large phased array antennas because acceptable radiation characteristics may be achieved with fewer elements, lower cost, smaller size, and weight and lower power consumption.

The design of a low-cost antenna array with 2D steering capability is very important for radiocommunication. As the satellite and mobile wireless communications technologies continue to utilize higher frequencies in the 20 to 60 GHz range, the design of a phased array antenna becomes prohibitively expensive. Also the realization od 2D scanning capability to track the satellite or to provide adequate urban communications systems becomes increasingly difficult to realize at a sufficiently low-cost suitable for the consumer market.

The aim of this paper is to present a new method of array pattern synthesis by taking into consideration pattern of each radiator specially of the ferroelectric one. Radiating elements employing ferroelectric materials may give much better performance compared with ferrite ones, because of their high power handling capability, low drive power, full military temperature range of operation and low cost. The main feature of the ferroelectric antennas is the change of ferroelectric material permittivity with an applied dc control voltage. The permittivity change by varying the dc bias enables to create different radiation patterns. This permits to use such radiating element in several applications, e.g. smart antenna arrays.

In this paper, new method of pattern synthesis of antenna arrays without phase shifters and attenuators has been presented. This concept is based on utilizing voltage-controlled ferroelectric array, where variable pattern of each antenna element is used to synthesize array pattrern.

Keywords: antenna arrays, pattern synthesis, voltage-controlled ferroelectric array

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2002, 48, z. 1, ss. 81–99

Projektowanie filtrów SOI o liniowej fazie z zastosowaniem aproksymacji równomiernie falistej w pasmie przepustowym i średniokwadratowej w pasmie zaporowym

FELICJA WYSOCKA-SCHILLAK

Instytut Telekomunikacji, Akademia Techniczno-Rolnicza
al. Prof. S. Kaliskiego 7, 85-796 Bydgoszcz
e-mail: felicja@mail.atr.bydgoszcz.pl

Otrzymano 2001.06.28
Autoryzowano 2001.10.16

W pracy przedstawiono metodę projektowania cyfrowych dolnoprzepustowych filtrów SOI o liniowej fazie i charakterystykach amplitudowych otrzymanych z zastosowaniem aproksymacji równomiernie falistej w pasmie przepustowym oraz średniokwadratowej w pasmie zaporowym. W metodzie tej zadanie zaprojektowania rozpatrywanego filtru jest przekształcane w zadanie optymalizacji dwukryterialnej, które przy odpowiednim sformułowaniu może być potraktowane jako zadanie optymalizacji jednokryterialnej z ograniczeniami. Rozpatrzono zadanie projektowania filtru SOI w dwóch przypadkach, a mianowicie z uwzględnieniem dodatkowego warunku dotyczącego maksymalnej dopuszczalnej amplitudy zafalowań w pasmie przepustowym oraz z uwzględnieniem warunku dotyczącego maksymalnej dopuszczalnej amplitudy w pasmie zaporowym. Przedstawiono również zastosowanie zaproponowanej metody do projektowania filtru Nyquista. Zaprezentowano ponadto cztery przykłady obliczeniowe.

Słowa kluczowe: filtry cyfrowe, filtry SOI, aproksymacja równomiernie falista, aproksymacja średniokwadratowa, optymalizacja

1. WPROWADZENIE

Przy projektowaniu filtrów cyfrowych o skończonej odpowiedzi impulsowej (w skrócie: SOI) najczęściej wykorzystywanymi rodzajami aproksymacji są: aproksymacja średniokwadratowa oraz aproksymacja minimaks, która w określonych warunkach jest równoważna aproksymacji równomiernie falistej [1-3]. Aproksymacja minimaks zadanej charakterystyki amplitudowej zapewnia zminimalizowanie maksymalnego odchylenia charakterystyki aproksymującej od charakterystyki aproksymowanej. Aproksymacja śre-

dniokwadratowa zapewnia natomiast zminimalizowanie energii błędu aproksymacji zadanej charakterystyki amplitudowej. Zasadniczą wadą aproksymacji średniokwadratowej jest jednak występowanie zjawiska Gibbsa [1-3], w wyniku czego w pobliżu punktów nieciągłości pojawiają się oscylacje o stosunkowo dużych amplitudach.

Aproksymacja równomiernie falista jest z reguły najwłaściwszym rodzajem aproksymacji, jaka może być zastosowana w pasmie przepustowym, gdyż istotne jest wówczas zminimalizowanie zniekształceń amplitudowych sygnału przechodzącego przez filtr. W pasmie zaporowym zastosowanie tego rodzaju aproksymacji nie jest jednak wskazane w tych przypadkach, gdy zależy nam na uzyskaniu możliwie małej wartości energii błędu aproksymacji charakterystyki amplitudowej. W przypadku filtrów o równomiernie falistych charakterystykach amplitudowych wartość tej energii może być bowiem stosunkowo duża. Jeżeli w danym zastosowaniu istotna jest mała wartość energii błędu aproksymacji charakterystyki amplitudowej w pasmie zaporowym, rozwiązaniem proponowanym przez Adamsa [4-5] jest przeprowadzenie równomiernie falistej aproksymacji charakterystyki amplitudowej w pasmie przepustowym oraz aproksymacji średniokwadratowej w pasmie zaporowym (w skrócie: aproksymacji RFSR).

W zadaniu zaprojektowania dolnoprzepustowego filtru SOI z zastosowaniem aproksymacji RFSR mamy do czynienia z aproksymacją w dwóch przedziałach rozłącznych, przy czym w każdym z tych przedziałów jest ona przeprowadzana w oparciu o różne kryteria. Oba kryteria powinny być uwzględniane jednocześnie w procesie projektowania filtru. Dodatkowo w zadaniu projektowania można jeszcze uwzględnić inne warunki, np. dotyczące maksymalnej dopuszczalnej amplitudy zafalowań δ_1 w pasmie przepustowym lub maksymalnej dopuszczalnej amplitudy δ_2 w pasmie zaporowym.

Zastosowanie zarówno aproksymacji równomiernie falistej jak i średniokwadratowej w procesie projektowania filtrów cyfrowych jest szeroko opisywane w literaturze. Istnieją jednak jedynie pojedyncze prace dotyczące rozwiązań kompromisowych pomiędzy aproksymacją równomiernie falistą i średniokwadratową [4-10].

W niniejszej pracy zaproponowano metodę projektowania dolnoprzepustowego filtru SOI o liniowej fazie z zastosowaniem aproksymacji RFSR. W metodzie tej zadanie zaprojektowania filtru jest przekształcane w odpowiednie zadanie polioptymalizacji z ograniczeniami. Zaproponowana metoda umożliwia uwzględnienie w procesie projektowania dodatkowych warunków w dziedzinie czasu i/lub w dziedzinie częstotliwości. W pracy przedstawiono rozwiązanie dwóch zadań projektowania dolnoprzepustowych filtrów SOI o liniowej fazie z zastosowaniem aproksymacji RFSR z uwzględnieniem dodatkowych warunków. W pierwszym z tych zadań występuje odpowiednio warunek dotyczący maksymalnej dopuszczalnej amplitudy zafalowań δ_1 w pasmie przepustowym lub maksymalnej dopuszczalnej amplitudy δ_2 w pasmie zaporowym. Jako drugie zadanie rozpatrywane jest projektowanie filtru Nyquista, w przypadku którego, oprócz warunku dotyczącego maksymalnej dopuszczalnej amplitudy zafalowań δ_1 w pasmie przepustowym, należy dodatkowo uwzględnić warunki dotyczące odpowiedzi impulsowej filtru. W pracy podano również cztery przykłady obliczeniowe ilustrujące wykorzystanie zaproponowanej metody.

2. SFORMUŁOWANIE PROBLEMU

2.1. PROJEKTOWANIE DOLNOPRZEPUSTOWEGO FILTRU SOI O LINIOWEJ FAZIE

Niech $h(n)$ będzie odpowiedzią impulsową jednowymiarowego filtru przyczynowego SOI o długości N. Charakterystyka częstotliwościowa (amplitudowo-fazowa) $H(e^{j\omega})$ takiego filtru jest określona zależnością [1]:

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n)e^{-j\omega n}, \tag{1}$$

gdzie: $\omega = 2\pi f$ – pulsacja unormowana względem częstotliwości próbkowania $F_p(\omega \in [0;\pi])$,
f – częstotliwość unormowana.

W przypadku filtrów SOI o liniowej fazie charakterystyka ta może być przedstawiona w następującej postaci [1]:

$$H(e^{j\omega}) = H(\omega)e^{j\phi(\omega)}, \tag{2}$$

gdzie $\phi(\omega) = \beta - \alpha\omega$, a $H(\omega)$ jest funkcją rzeczywistą, nazywaną w literaturze charakterystyką częstotliwościową o fazie zerowej (ang. *zero-phase frequency response*) [1].

Rozpatrzmy teraz dolnoprzepustowy filtr cyfrowy SOI o liniowej fazie i symetrycznej odpowiedzi impulsowej, której długość N jest liczbą nieparzystą. Charakterystyka częstotliwościowa o fazie zerowej $H(\omega)$ takiego filtru jest wyrażona następująco [1-3]:

$$H(\omega) = h(0) + 2\sum_{n=1}^{M} h(n)\cos(\omega n), \tag{3}$$

gdzie: $M = (N-1)/2$.

Charakterystyka amplitudowa $A(\omega)$ filtru jest określona wzorem:

$$A(\omega) = |H(e^{j\omega})| = |H(\omega)|. \tag{4}$$

Zadanie zaprojektowania filtru o określonej charakterystyce amplitudowej, sprowadza się więc do wyznaczenia odpowiednich wartości współczynników $h(n)$, $n = 0,1,...,M$.

W rozpatrywanym przez nas przypadku żądamy, aby charakterystyka częstotliwościowa o fazie zerowej projektowanego filtru aproksymowała charakterystykę częstotliwościową idealnego filtru dolnoprzepustowego o postaci

$$H_d(e^{j\omega}) = \begin{cases} 1 & , \quad 0 \le \omega \le \omega p, \\ 0 & , \quad \omega_s \le \omega \le \pi, \end{cases} \tag{5}$$

gdzie: ω_p i ω_s – unormowane pulsacje graniczne odpowiednio pasma przepustowego i zaporowego.

Oznaczmy przez $\mathbf{Y} = [y_1, y_2, ..., y_{M+1}]^T$ wektor o współrzędnych zdefiniowanych następująco:

$$y_i = h(i-1), \quad i = 1,2,...M+1, \tag{6}$$

a przez $H(\omega,\mathbf{Y})$ charakterystykę częstotliwościową o fazie zerowej otrzymaną przy przyjęciu we wzorze (3) współczynników filtru o wartościach określonych przez wektor $\mathbf{Y}$.

Funkcja błędu aproksymacji $E(\omega,\mathbf{Y})$ charakterystyki częstotliwościowej jest wyrażona ogólnym wzorem:

$$E(\omega,\mathbf{Y}) = [H(\omega,\mathbf{Y}) - H_d(e^{j\omega})]. \tag{7}$$

Wzór ten może być również zastosowany w przypadku równomiernie falistej aproksymacji charakterystyki częstotliwościowej o fazie zerowej w pasmie przepustowym $P = [0,\omega_p]$. W pasmie zaporowym $S = [\omega_s, \pi]$ należy natomiast zminimalizować błąd średniokwadratowy E_2 określony zależnością:

$$E_2 = \int_{\omega_s}^{\pi} [H(\omega,\mathbf{Y}) - H_d(e^{j\omega})]^2 d\omega = \int_{\omega_s}^{\pi} [H(\omega,\mathbf{Y})]^2 d\omega, \tag{8}$$

Zadanie projektowania filtru dolnoprzepustowego sformułujemy w sposób następujący: Dla zadanych wartości M, ω_p, ω_s oraz δ_1 znaleźć wektor $\mathbf{Y} = [y_1, y_2, ..., y_{M+1}]^T$, dla którego funkcja błędu $E(\omega, \mathbf{Y})$ ma przebieg równomiernie falisty w pasmie przepustowym, a w pasmie zaporowym zminimalizowany jest błąd średniokwadratowy określony wzorem (8), przy dodatkowym spełnieniu jednego z następujących warunków: warunku dotyczącego maksymalnej dopuszczalnej amplitudy zafalowań w pasmie przepustowym:

$$\forall_{\omega \in P} \left| H(\omega,\mathbf{Y}) - H_d(e^{j\omega}) \right| \le \delta_1, \tag{9}$$

lub warunku dotyczącego maksymalnej dopuszczalnej amplitudy w pasmie zaporowym:

$$\forall_{\omega \in S} \left| H(\omega,\mathbf{Y}) \right| \le \delta_2 \tag{10}$$

2.2. PROJEKTOWANIE FILTRU NYQUISTA SOI O LINIOWEJ FAZIE

Cyfrowe filtry Nyquista są z reguły realizowane jako filtry SOI o liniowej fazie. Odpowiedź impulsowa $h(n)$ filtru Nyquista przyjmuje wartość zero co n próbek poczynając od próbki centralnej, gdzie n jest liczbą całkowitą. Zastosowanie tych filtrów

umożliwia otrzymanie widma sygnału zawartego w ściśle ograniczonym pasmie częstotliwości oraz wyeliminowanie interferencji międzysymbolowej [1, 11]. Filtry te mogą być również wykorzystywane jako filtry interpolacyjne i decymacyjne [1, 12].

Zaprojektowanie filtru SOI o liniowej fazie sprowadza się do wyznaczenia jego odpowiedzi impulsowej w taki sposób, aby charakterystyka amplitudowa filtru spełniała określone wymagania. W przypadku filtrów Nyquista, oprócz wymagań dotyczących charakterystyki amplitudowej należy również uwzględnić wymagania dotyczące odpowiedzi impulsowej filtru.

Zgodnie z twierdzeniem Nyquista, dysponując pasmem o szerokości f_N herców można przesłać $2f_N$ niezależnych próbek sygnału na sekundę. Taki przypadek określany jest jako transmisja z szybkością Nyquista bez interferencji międzysymbolowej. Warunek konieczny i dostateczny wyeliminowania interferencji międzysymbolowej nosi w literaturze nazwę pierwszego kryterium Nyquista. W przypadku cyfrowego filtru SOI warunek ten może być wyrażony w następującej postaci [1, 13-14]:

$$h(n) = \begin{cases} C - \text{dla } n = n_0 \\ 0 - \text{dla } n = n_0 + kL, \quad k = 1,2,3... \end{cases}, \tag{11}$$

gdzie: $C \neq 0$ – stała,
n_0 – liczba całkowita dodatnia,
$L = F_P/R$ – liczba całkowita dodatnia,
F_P – częstotliwość próbkowania,
R – szybkość Nyquista w bitach na sekundę.

Filtr, którego odpowiedź impulsowa spełnia powyższe warunki nazywany jest filtrem Nyquista. Jeżeli filtr ten jest realizowany jako filtr SOI o liniowej fazie, jego odpowiedź impulsowa jest symetryczna względem osi pionowej przechodzącej przez punkt $n = n_0$.

Szczególnym przypadkiem filtrów Nyquista są tzw. filtry półpasmowe [1, 12]. W przypadku tych filtrów $L = 2$, czyli co druga próbka odpowiedzi impulsowej filtru przyjmuje wartość zero.

Charakterystyka częstotliwościowa idealnego filtru Nyquista jest określona następująco [14]:

$$H_d(e^{j\omega}) = \begin{cases} 1 \ , & 0 \leq \omega < \pi/L \ , \\ 0 \ , & \pi/L < \omega \leq \pi \ . \end{cases} \tag{12}$$

Jak wiadomo, filtr idealny jest nierealizowalny fizycznie i w praktyce są stosowane filtry o charakterystykach aproksymujących charakterystykę częstotliwościową filtru idealnego.

W przypadku charakterystyki częstotliwościowej rzeczywistego filtru Nyquista definiuje się pulsacje krańcowe odpowiednio pasma przepustowego i zaporowego w sposób następujący [1, 13-14]:

$$\omega_p = \frac{(1-\alpha)}{L}\pi, \tag{13}$$

$$\omega_s = \frac{(1+\alpha)}{L}\pi, \tag{14}$$

gdzie: α – parametr z przedziału $\alpha \in [0,1]$.

Po uwzględnieniu warunków wynikających z założenia, że rozpatrywany filtr ma być filtrem Nyquista, charakterystyka częstotliwościowa o fazie zerowej dolnoprzepustowego filtru o symetrycznej odpowiedzi impulsowej i nieparzystej długości N przyjmuje postać [14]:

$$H_N(\omega) = 1/L + 2\sum_{\substack{n=1 \\ n\notin Z}}^{M} h(n)\cos(\omega n), \tag{15}$$

gdzie Z jest zbiorem wskaźników n, dla których współczynniki $h(n)$ są równe zeru. Liczba wskaźników n należących do zbioru Z wynosi $E[M/L]$, gdzie $E[x]$ oznacza część całkowitą (entier) liczby x.

Zadanie zaprojektowania filtru Nyquista sprowadza się do wyznaczenia odpowiednich wartości współczynników $h(n)$, $n = 1,2,...,M$ i $n \notin Z$. Liczba I tych współczynników jest określona zależnością [14]:

$$I = (N-1)/2 - E[N/(2L)]. \tag{16}$$

Zadanie zaprojektowania filtru Nyquista sformułujemy w sposób następujący:
Dla zadanych wartości N, L, α oraz δ_1 znaleźć wektor $\mathbf{Y} = [y_1 y_2,...,y_{M+1}]^T$o współrzędnych zdefiniowanych wzorem (6), dla którego funkcja błędu $E(\omega,\mathbf{Y})$ ma przebieg równomiernie falisty w pasmie przepustowym, a w pasmie zaporowym zminimalizowany jest błąd średniokwadratowy określony wzorem (8), przy spełnieniu warunku (9) dotyczącego maksymalnej dopuszczalnej amplitudy zafalowań w pasmie przepustowym oraz następujących warunkach ograniczających:

$$y_1 = C = 1/L, \tag{17}$$

$$y_k = 0 \quad \text{dla} \quad k = L, 2L, \ldots, E[N/(2L)]L. \tag{18}$$

Należy zauważyć, że w porównaniu z zadaniem projektowania filtru SOI o liniowej fazie sformułowanym w p. 2.1., w zadaniu powyższym występują dwa dodatkowe liniowe warunki ograniczające (17) i (18), których spełnienie zapewnia, że otrzymany filtr jest filtrem Nyquista.

3. PRZEKSZTAŁCENIE PROBLEMU W ZADANIE OPTYMALIZACJI DWUKRYTERIALNEJ

Sformułowane w p. 2.1. oraz 2.2. zadania projektowania filtrów SOI można przekształcić w odpowiednie zadanie optymalizacji dwukryterialnej z ograniczeniami. W tym celu należy ustalić dwa wskaźniki jakości odpowiadające dwom kryteriom występującym w zadaniu projektowania filtru, odpowiednio dla pasma przepustowego i zaporowego. Określenie postaci pierwszego z tych wskaźników rozpocznijmy od podzielenia, przy założonych wartościach początkowych wektora $\mathbf{Y}$, pasma przepustowego $P = [0,\omega_p]$ na przedziały θ_j, $j = 1,2,...,J$, zdefiniowane w taki sposób, że wewnątrz każdego z tych przedziałów funkcja błędu ma dokładnie jedno ekstremum. Niech przedziały te będą określone w sposób następujący:

$$\theta_1 = [0,\ \omega_1),$$

$$\theta_j = [\omega_{j-1},\ \omega_j),\quad j = 2,3,\ldots,J-1, \tag{19}$$

$$\theta_J = [\omega_{J-1},\ \omega_p],$$

gdzie: J jest liczbą przedziałów θ_j występujących w pasmie przepustowym, a ω_j, $j = 1,2,...,\ J-1$, $(\omega_1 < \omega_2 < ... < \omega_{J-1})$ są unormowanymi pulsacjami, dla których spełnione są warunki:

$$(-1)^{m+j}\frac{dH(\omega,\mathbf{Y})}{d\omega}\bigg|_{\omega=\omega_j} < 0,\quad j = 1,2,...,J-1, \tag{20}$$

gdzie:

$$m = \begin{cases} 0 - \text{gdy dla pulsacji } \omega = 0 \text{ istnieje maksimum lokalne,} \\ 1 - \text{gdy dla pulsacji } \omega = 0 \text{ istnieje minimum lokalne.} \end{cases}$$

W celu określenia wartości ekstremów funkcji błędu $E(\omega,\mathbf{Y})$ wyznaczmy następnie wartości największych odległości $\Delta E_i(\mathbf{Y})$, $i = 1,2,...,J+2$ pomiędzy charakterystykami $H_d(e^{j\omega})$ i $H(\omega,\mathbf{Y})$ w poszczególnych przedziałach θ_j, $j = 1,2,...,J$, oraz wartości odległości pomiędzy tymi charakterystykami dla pulsacji $\omega = 0$ i dla pulsacji granicznej pasma przepustowego $\omega = \omega_p$ w sposób następujący:
– w przypadku, gdy dla pulsacji $\omega = 0$ występuje minimum lokalne

$$\Delta E_1(\mathbf{Y}) = H_d(e^{j0}) - H(0,\mathbf{Y}); \tag{21}$$

$$\Delta E_i(\mathbf{Y}) = \max_{\omega\in\theta_{i-1}}(H(\omega,\mathbf{Y}) - H_d(e^{j\omega})),\quad i = 2,4,6,...,\quad (i \le J+1); \tag{22}$$

$$\Delta E_i(\mathbf{Y}) = \max_{\omega \in \theta_{i-1}} (H_d(e^{j\omega}) - \mathrm{H}(\omega,\mathbf{Y})), \quad i = 3,5,7,..., \quad (i \le J+1); \tag{23}$$

$$\Delta E_{J+2}(\mathbf{Y}) = H(\omega_p,\mathbf{Y}) - H_d(e^{j\omega_p}); \tag{24}$$

– w przypadku, gdy dla pulsacji $\omega = 0$ występuje maksimum lokalne

$$\Delta E_1(\mathbf{Y}) = H(0,\mathbf{Y}) - H_d(e^{j0}); \tag{25}$$

$$\Delta E_i(\mathbf{Y}) = \max_{\omega \in \theta_{i-1}} (H_d(e^{j\omega}) - \mathrm{H}(\omega, \mathbf{Y})), \quad i = 2,4,6,..., \quad (i \le J+1); \tag{26}$$

$$\Delta E_i(\mathbf{Y}) = \max_{\omega \in \theta_{i-1}} (H(\omega,\mathbf{Y}) - H_d(e^{j\omega})), \quad i = 3,5,7,..., \quad (i \le J+1); \tag{27}$$

$$\Delta E_{J+2}(\mathbf{Y}) = H_d(e^{j\omega_p}) - H(\omega_p, \mathbf{Y}). \tag{28}$$

Zadanie zaprojektowania filtru o równomiernie falistej charakterystyce amplitudowej w pasmie przepustowym sprowadza się do znalezienia takiego wektora $\mathbf{Y} = [y_1, y_2,...,y_{M+1}]^T$, dla którego spełniony jest warunek

$$\Delta E_i(\mathbf{Y}) = \Delta E_k(\mathbf{Y}), \; k,i = 1,2,...,J+2 \tag{29}$$

W celu wyznaczenia wektora $\mathbf{Y}$ spełniającego powyższy warunek, wprowadźmy funkcję $X_1(\mathbf{Y})$ określoną następująco:

$$X_1(\mathbf{Y}) = \sum_{i=1}^{J+2} \left(\Delta E_i(\mathbf{Y}) - \frac{1}{J+2} \sum_{j=1}^{J+2} \Delta E_j(\mathbf{Y}) \right)^2 \tag{30}$$

Tak określona funkcja $X_1(\mathbf{Y})$:

- jest nieujemną funkcją ciągłą zmiennych ΔE_1, ΔE_2, ..., ΔE_{J+2},
- przyjmuje wartość zero wtedy i tyko wtedy, gdy $\Delta E_1 = \Delta E_2 = ... = \Delta E_{J+2}$,
- nie posiada lokalnych minimów.

Ponieważ wartości zmiennych ΔE_1, ΔE_2,..., ΔE_{J+2} są zależne od wartości współczynników wektora $\mathbf{Y}$, funkcję X_1 możemy traktować jako funkcję zmiennych y_1, $y_2,...,y_{M+1}$. Funkcja X_1 zdefiniowana wzorem (30) może być więc zastosowana jako pierwszy ze wskaźników jakości występujących w zadaniu optymalizacji dwukryterialnej. Jako drugi wskaźnik jakości zostanie użyty błąd E_2 określony zależnością (8).

Mając tak ustalone dwa wskaźniki jakości, równoważne zadanie optymalizacji dwukryterialnej z ograniczeniami sformułujemy w sposób następujący:
Dla przyjętych wartości współczynników wagowych β_1 oraz β_2 znaleźć wektor $\mathbf{Y} = [y_1,y_2,...,y_{M+1}]^T$, który minimalizuje syntetyczny wskaźnik jakości $X(\mathbf{Y},\beta_1,\beta_2)$

$$X(\mathbf{Y},\beta_1,\beta_2) = \beta_1 \sum_{i=1}^{J+2} (\Delta E_i(\mathbf{Y}) - \tilde{S})^2 + \beta_2 \int_{\omega_s}^{\pi} [H(\omega,\mathbf{Y})]^2 d\omega, \tag{31}$$

przy czym

$$\tilde{S} = \frac{1}{J+2} \sum_{i=1}^{J+2} \Delta E_i(\mathbf{Y}) \tag{32}$$

przy warunkach ograniczających:

$$\Delta E_l(\mathbf{Y}) > 0, \quad l = 1,2,...,J+2, \tag{33}$$

$$\forall_{\omega\in(\omega_p,\omega_s)} \; H(\omega,\mathbf{Y}) - 1 < 0 \tag{34}$$

oraz warunku (9):

$$\forall_{\omega\in P} \; |H(\omega,\mathbf{Y}) - H_d(e^{j\omega})| \le \delta_1$$

lub (10):

$$\forall_{\omega\in S} \; |H(\omega,\mathbf{Y})| \le \delta_2.$$

W przypadku projektowania filtru Nyquista należy w powyższym zadaniu dodatkowo uwzględnić warunki (17) i (18).

Należy zauważyć, że spełnienie ograniczenia (34) zapewnia kontrolę przebiegu funkcji aproksymującej $H(\omega, \mathbf{Y})$ w przedziale (ω_p, ω_s), tak aby nie przyjmowała ona wartości większych od 1.

Sformułowane powyżej zadanie optymalizacji dwukryterialnej może być potraktowane jako zadanie optymalizacji jednokryterialnej z ograniczeniami o wskaźniku jakości danym wzorem (31). Zadanie tego typu można rozwiązać jedną z metod poszukiwania minimum z ograniczeniami. Otrzymane rozwiązanie jest funkcją przyjętych wartości współczynników wagowych β_1 oraz β_2. Wartości tych współczynników są przyjmowane w zależności od założonych priorytetów. Jeżeli chcemy uzyskać równomiernie falistą charakterystykę amplitudową filtru w pasmie przepustowym, a jednocześnie w pasmie zaporowym ma być zminimalizowany błąd średniokwadratowy określony wzorem (8), należy tak dobrać wartości β_1 oraz β_2, aby początkowe wartości obu składników wskaźnika jakości (31) były w przybliżeniu takie same. Jeżeli bardziej zależy nam na uzyskaniu dokładnie równomiernie falistej charakterystyki amplitudowej w pasmie przepustowym, niż na dokładnej minimalizacji błędu średniokwadratowego

w pasmie zaporowym, należy przyjąć takie wartości β_1 i β_2, aby początkowa wartość pierwszego składnika we wzorze (31) była wyraźnie, np. o jeden rząd wielkości, większa niż drugiego. Jeżeli natomiast bardziej zależy nam na dokładnej minimalizacji błędu średniokwadratowego w pasmie zaporowym, należy postąpić odwrotnie, tzn. tak dobrać β_1 i β_2, aby początkowa wartość drugiego składnika we wzorze (31) była większa niż pierwszego.

Z podanych w literaturze [15, 16] metod, do rozwiązywania rozpatrywanego zadania wybrano metodę przesuwanej funkcji kary. Metoda ta uważana jest bowiem za jedną z najbardziej efektywnych metod poszukiwania minimum z ograniczeniami. Umożliwia ona sprowadzenie zadania z ograniczeniami do ciągu zadań bez ograniczeń poprzez odpowiednią modyfikację funkcji celu. Zadania programowania nieliniowego bez ograniczeń rozwiązywano metodą gradientu sprzężonego Polaka-Ribiery [15-17].

Przy rozwiązywaniu zadań optymalizacji, podobnie jak przy rozwiązywaniu innych zagadnień metodami numerycznymi, istotnym problemem jest wybór odpowiedniego punktu startowego. Punkt ten powinien być w ogólnym przypadku niezbyt odległy od poszukiwanego rozwiązania, gdyż przyjęcie zbyt odległego punktu startowego może niekiedy powodować numeryczną rozbieżność algorytmu.

Początkowe wartości współrzędnych wektora **Y**, służące jako punkt startowy przy rozwiązywaniu rozpatrywanego zadania optymalizacji, wyznaczono przybliżając funkcję $H_{d0}(e^{j\omega})$ określoną zależnością:

$$H_{d0}(e^{j\omega}) = \begin{cases} 1 & , \quad 0 \le \omega \le \omega_p \\ \dfrac{\omega - \omega_s}{\omega_p - \omega_s} & , \quad \omega_p < \omega < \omega_s \\ 0 & , \quad \omega_s \le \omega \le \pi \end{cases} \tag{32}$$

sumą $M+1$ początkowych wyrazów (od $h(0)$ do $h(M)$) cosinusowego szeregu Fouriera. Współczynniki tego szeregu są wyrażone następująco:

$$h_0 = h(0) = \frac{1}{\pi}\int_0^{\pi} H_{d0}(e^{j\omega})d\omega = \frac{1}{2\pi}(\omega_p + \omega_s), \tag{33}$$

$$h_k = h(k) = \frac{2}{\pi}\int_0^{\pi} H_{d0}(e^{j\omega})\cos(k\omega)d\omega = \tag{34}$$

$$= \frac{-4}{k^2\pi(\omega_s - \omega_p)} \sin(k\frac{\omega_s + \omega_p}{2})\sin(k\frac{\omega_s - \omega_p}{2}), \quad k = 1,2,3...$$

Przyjęcie takich współczynników powoduje wyraźne zmniejszenie amplitud oscylacji funkcji $H(\omega,\mathbf{Y})$ w pobliżu pulsacji ω_p i ω_s w porównaniu z przypadkiem zastosowania współczynników szeregu Fouriera aproksymującego funkcję $H_d(e^{j\omega})$ daną wzorem (5). Tak określony punkt startowy jest najlepszą aproksymacją funkcji $H_{d0}(e^{j\omega})$

w sensie normy L_2. Jest on bliski poszukiwanego rozwiązania, a jego użycie w procedurze minimalizacyjnej powoduje skrócenie czasu wykonywania obliczeń w porównaniu z przypadkiem zastosowania jako punktu startowego współczynników szeregu Fouriera aproksymującego funkcję $H_d(e^{j\omega})$. Ponadto, w przypadku tak wybranego punktu startowego charakterystyka $H(\omega,\mathbf{Y})$ ma występujące na przemian po sobie minima i maksima lokalne w pasmie przepustowym, co jest niezbędne do wyznaczenia przedziałów, θ_j, $j = 1,2,...,J$.

4. PRZYKŁADY OBLICZENIOWE

Na podstawie przedstawionej metody projektowania filtrów SOI o liniowej fazie z zastosowaniem aproksymacji RFSR zostały opracowane dwa programy komputerowe EQLS oraz EQLSNY. Program EQLS umożliwia projektowanie dolnoprzepustowych filtrów SOI o liniowej fazie z uwzględnieniem jednego z dwóch warunków: warunku dotyczącego maksymalnej dopuszczalnej amplitudy zafalowań w pasmie przepustowym lub warunku dotyczącego maksymalnej dopuszczalnej amplitudy w pasmie zaporowym. Program EQLSNY umożliwia natomiast projektowanie filtrów Nyquista. Oba te programy ten zostały napisane w języku Fortran 77, przy czym do kompilacji wykorzystano kompilator PowerStation firmy Microsoft (wersja 4.0).

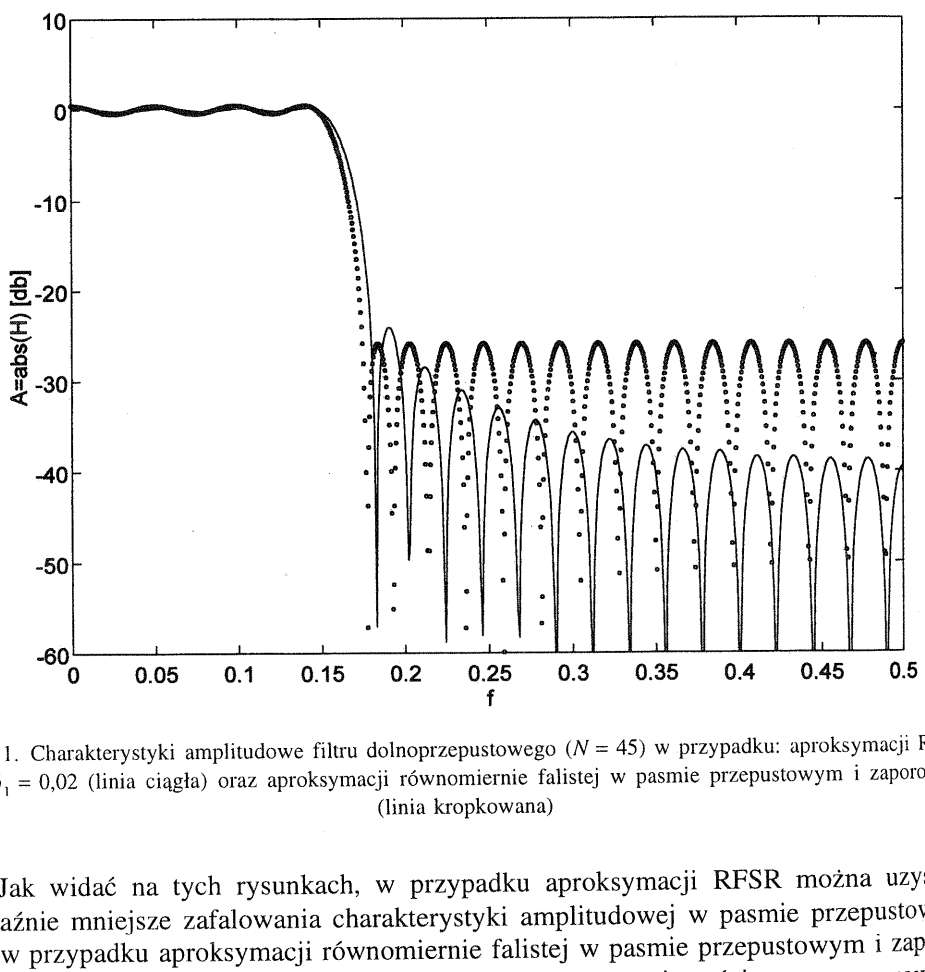
W obu opracowanych programach do obliczenia wartości funkcji celu $X_1(\mathbf{Y})$ konieczne jest określenie wartości największych odległości $\Delta E_i(\mathbf{Y})$, $i = 1,2,...,J+2$ pomiędzy charakterystykami $H_d(e^{j\omega})$ i $H(\omega,\mathbf{Y})$ w poszczególnych przedziałach θ_j, $j = 1,2,...,J$. Wartości ekstremów funkcji $H(\omega,\mathbf{Y})$ w przedziałach θ_j wyznaczono posługując się metodą złotego podziału [15-17]. Metoda ta umożliwia znalezienie odciętej a oraz rzędnej $f(a)$ minimum rozpatrywanej funkcji $f(a)$ w określonym przedziale, przy założeniu, że w tym przedziale funkcja $f(a)$ jest ciągła i ma dokładnie jedno minimum.

Do rozwiązywania zadań optymalizacji bez ograniczeń w obu programach wykorzystano odpowiednio zmodyfikowaną procedurę GSPR1E opracowaną w Instytucie Automatyki Politechniki Warszawskiej [16]. Procedura ta umożliwia rozwiązywanie zadań programowania nieliniowego bez ograniczeń metodą gradientu sprzężonego Polaka-Ribiery, przy czym gradient funkcji celu jest estymowany numerycznie.

Przy użyciu obu programów przeprowadzono projektowanie wielu dolnoprzepustowych filtrów SOI z zastosowaniem aproksymacji RFSR. Obliczenia wykonywano przy użyciu mikrokomputera IBM PC Pentium z zegarem 450 MHz. Czas wykonywania obliczeń w przypadku filtrów o długości odpowiedzi impulsowej N mniejszej od 50 wahał się od dziesiątych części sekundy do pojedynczych sekund, przy czym czas ten wyraźnie zwiększał się w miarę wzrostu liczby N.

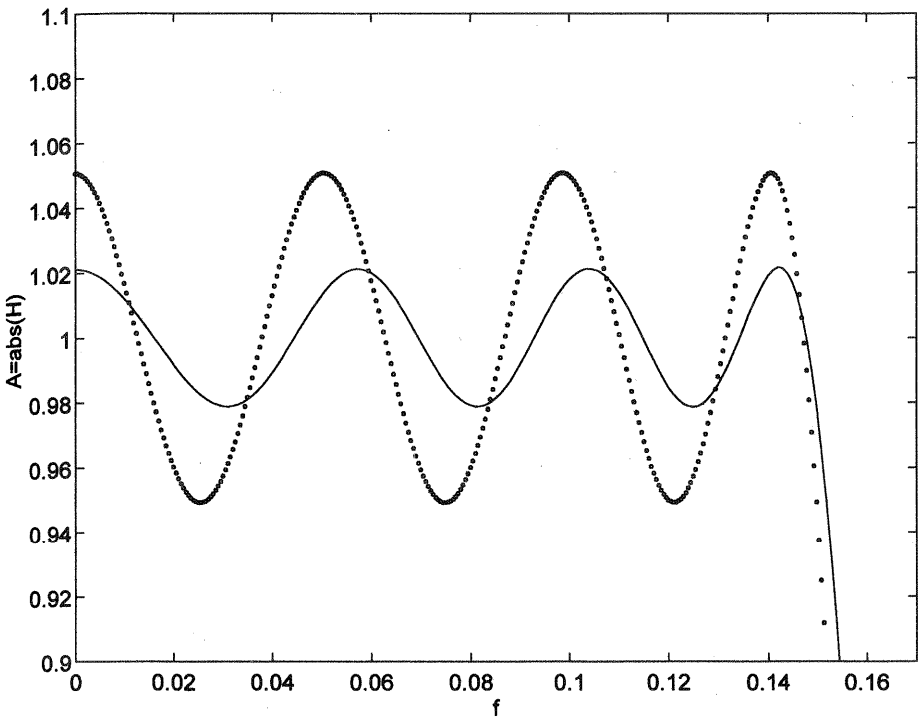
W celu zademonstrowania przykładowych wyników otrzymanych przy użyciu programu EQLS z uwzględnieniem warunku dotyczącego maksymalnej dopuszczalnej amplitudy zafalowań w pasmie przepustowym zaprojektowano dolnoprzepustowy filtr SOI o liniowej fazie dla następujących danych: $N = 45$, $f_p = \omega_p/(2\pi) = 0{,}15$, $f_s = \omega_s/(2\pi) = 0{,}175$ oraz $\delta_1 = 0{,}02$. Do obliczeń przyjęto wartość parametru $\beta_1 = 20$

oraz $\beta_2 = 1$. Aby porównać otrzymaną charakterystykę amplitudową z charakterystyką amplitudową równomiernie falistą w pasmie przepustowym i zaporowym, zaprojektowano następnie odpowiedni filtr dla takich samych wartości N, f_p, f_s oraz wagi $w = 1$, przy użyciu funkcji *remez* występującej w *toolboxie* „Signal Processing'' pakietu programów MATLAB [18]. Funkcja *remez* jest implementacją algorytmu Parksa-McClellana [1, 12] umożliwiającego projektowanie filtrów SOI o liniowej fazie i równomiernie falistych charakterystykach amplitudowych w pasmie przepustowym i zaporowym. Przebiegi wyznaczonych charakterystyk amplitudowych są pokazane odpowiednio na rysunkach 1 i 2.



Rys. 1. Charakterystyki amplitudowe filtru dolnoprzepustowego ($N = 45$) w przypadku: aproksymacji RFSR dla $\delta_1 = 0{,}02$ (linia ciągła) oraz aproksymacji równomiernie falistej w pasmie przepustowym i zaporowym (linia kropkowana)

Jak widać na tych rysunkach, w przypadku aproksymacji RFSR można uzyskać wyraźnie mniejsze zafalowania charakterystyki amplitudowej w pasmie przepustowym niż w przypadku aproksymacji równomiernie falistej w pasmie przepustowym i zaporowym. Ponadto, w przypadku aproksymacji RFSR w znacznej części pasma zaporowego uzyskuje się tłumienie o ponad 10 dB większe niż w przypadku aproksymacji równomiernie falistej w obu pasmach. Jedynie tłumienie w pasmie zaporowym przy częstotliwościach bliskich f_s jest w przypadku aproksymacji RFSR nieco mniejsze niż w przypadku aproksymacji równomiernie falistej w pasmie przepustowym i zaporowym.



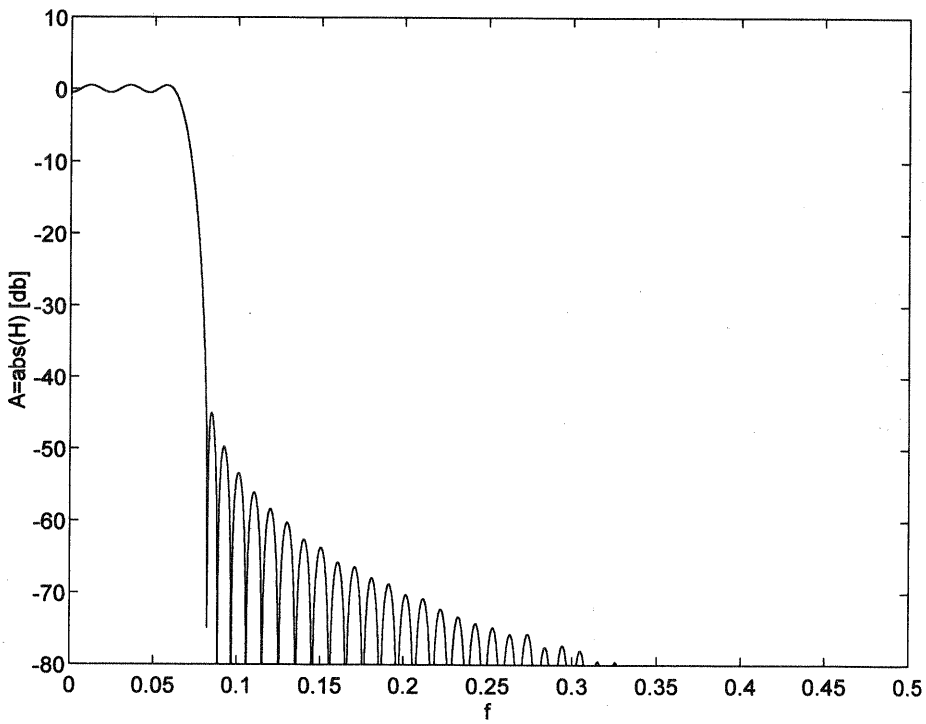
Rys. 2. Charakterystyki amplitudowe z rys. 1 rozciągnięte w pasmie przepustowym w przypadku: aproksymacji RFSR dla $\delta_1 = 0{,}02$ (linia ciągła) oraz aproksymacji równomiernie falistej w pasmie przepustowym i zaporowym (linia kropkowana)

W celu porównania wyników otrzymanych przy użyciu programu EQLS z uwzględnieniem warunku dotyczącego maksymalnej dopuszczalnej amplitudy zafalowań w pasmie przepustowym z wynikami uzyskanymi przez Adamsa, przeliczono przy użyciu programu EQLS przykład obliczeniowy zamieszczony w [4]. Zaprojektowano mianowicie filtr dolnoprzepustowy dla następujących danych: $N = 95$, $f_p = 0{,}0625$, $f_s = 0{,}0804$ oraz $\delta_1 = 0{,}0575$[1]. Do obliczeń przyjęto wartość parametru $\beta_1 = 1$ oraz $\beta_2 = 1$. Przebieg wyznaczonej charakterystyki amplitudowej jest pokazany na rysunku 3. Charakterystyka ta praktycznie nie różni się od odpowiedniej charakterystyki zamieszczonej w [4]. Na podstawie otrzymanych wyników obliczono również podaną przez Adamsa wartość DB_s, określającą stosunek maksymalnego zafalowania δ_2 charakterystyki amplitudowej w paśmie zaporowym do maksymalnego wzmocnienia w pasmie przepustowym, zdefiniowaną wzorem:

$$DB_s = 20\log_{10}\frac{\delta_s}{1+\delta_p} \quad [\text{dB}] \tag{35}$$

[1] W pracy Adamsa jest podana wartość zakładanego zafalowania międzyszczytowego $DB_P = 1{,}0$ dB, co odpowiada wartości $\delta_1 = 0{,}0575$.

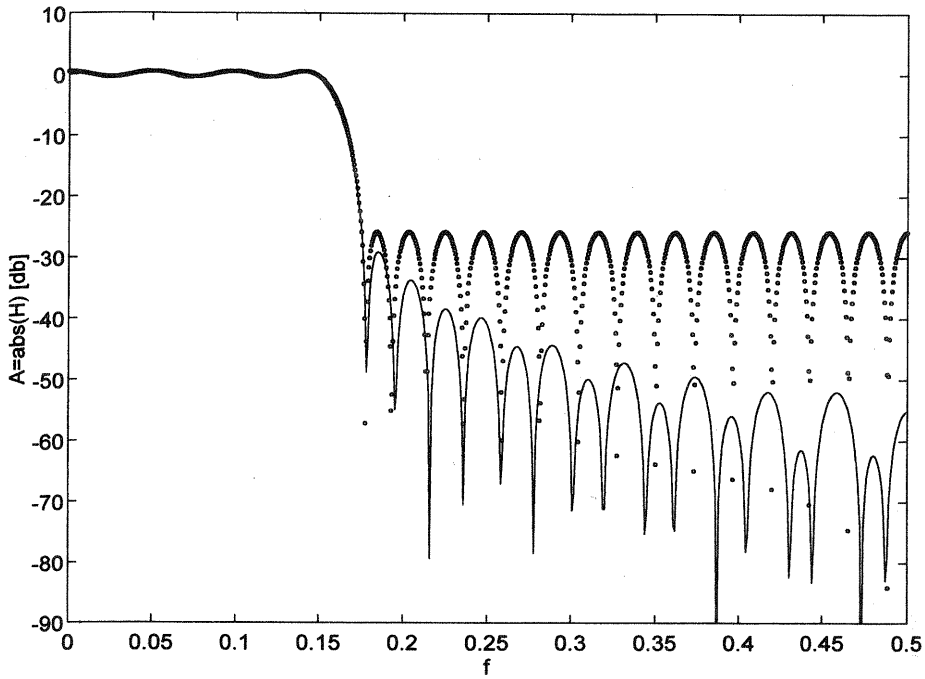
W przykładzie przeliczonym metodą Adamsa $DB_s = -36.19$ dB, natomiast w przypadku obliczeń wykonanych zaproponowaną metodą $DB_s = -36.72$ dB. Można więc przyjąć że otrzymane wyniki są praktycznie identyczne. Zaproponowana metoda umożliwia więc odtworzenie wyników uzyskanych metodą Adamsa.



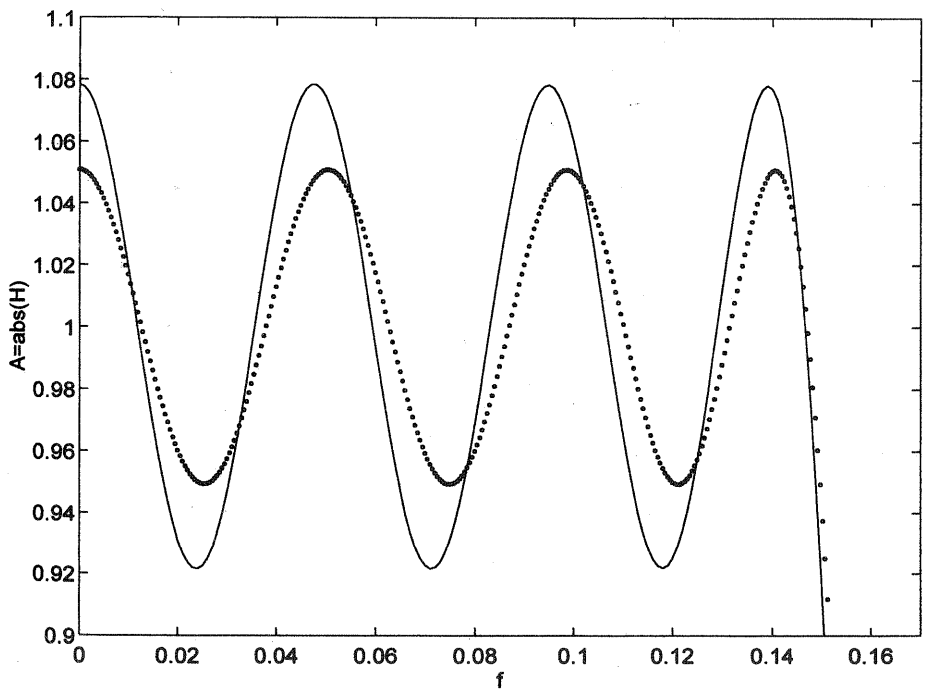
Rys. 3. Charakterystyka amplitudowa filtru dolnoprzepustowego ($N = 95$) z przykładu zamieszczonego w pracy Adamsa [4]

Rozpatrzmy teraz przypadek aproksymacji RFSR z uwzględnieniem warunku dotyczącego maksymalnej dopuszczalnej amplitudy w pasmie zaporowym. W celu zademonstrowania przykładowych wyników otrzymanych przy użyciu programu EQLS, zaprojektowano filtr dolnoprzepustowy dla następujących danych: $N = 45$, $f_p = 0{,}15$, $f_s = 0{,}175$, $\delta_2 = 0{,}05$. Do obliczeń przyjęto wartość parametru $\beta_1 = 2$ oraz $\beta_2 = 1$. Przebiegi wyznaczonej charakterystyki amplitudowej oraz charakterystyki amplitudowej otrzymanej przy zastosowaniu aproksymacji równomiernie falistej w pasmie przepustowym i zaporowym są pokazane odpowiednio na rysunkach 4 i 5.

Jak widać na tych rysunkach, w przypadku aproksymacji RFSR można uzyskać w znacznej części pasma zaporowego tłumienie o ponad 20 dB większe niż w przypadku aproksymacji równomiernie falistej w pasmie przepustowym i zaporowym. Odbywa się to jednak kosztem zwiększenia zafalowań charakterystyki amplitudowej w pasmie przepustowym.

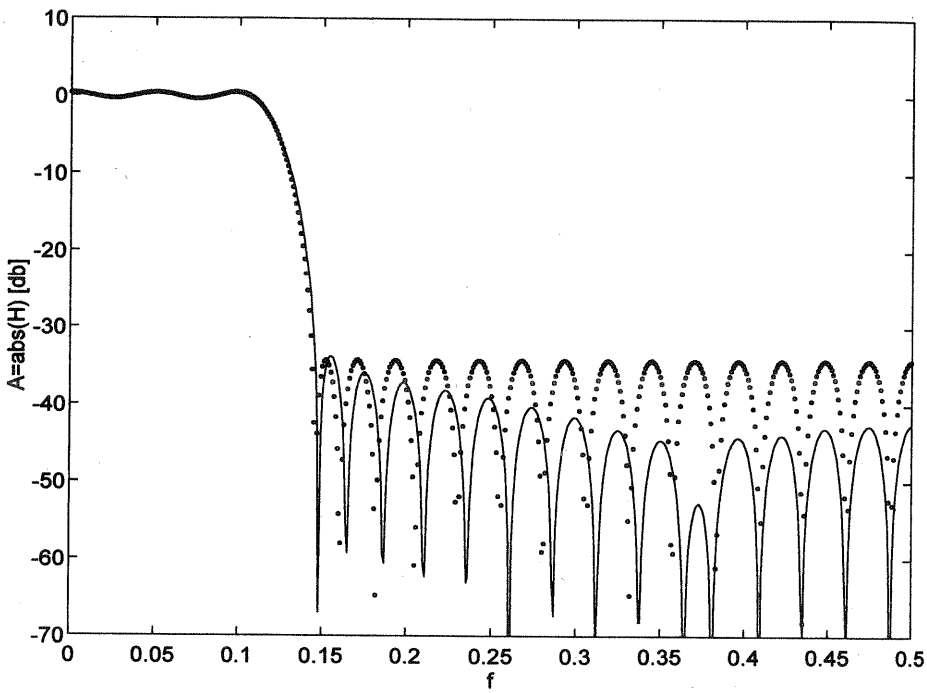


Rys. 4. Charakterystyka amplitudowa filtru dolnoprzepustowego ($N = 45$) w przypadku: aproksymacji RFSR dla $\delta_2 = 0{,}05$ (linia ciągła) aproksymacji równomiernie falistej w pasmie przepustowym i zaporowym (linia kropkowana)



Rys. 5. Charakterystyki amplitudowe z rys. 4 rozciągnięte w pasmie przepustowym w przypadku: aproksymacji RFSR dla $\delta_2 = 0{,}05$ (linia ciągła) oraz aproksymacji równomiernie falistej w pasmie przepustowym i zaporowym (linia kropkowana)

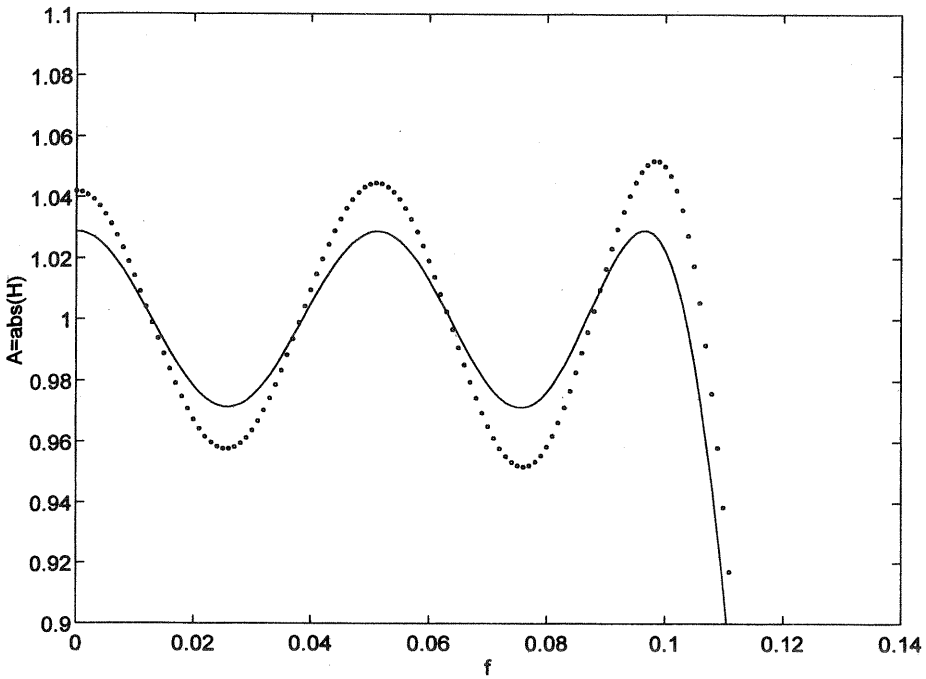
W celu przedstawienia przykładowych wyników otrzymanych przy użyciu programu EQLSNY, zaprojektowano filtr Nyquista dla następujących danych: $N = 39$, $L = 4$, $\alpha = 0{,}15$ oraz $\delta_1 = 0{,}022$. Do obliczeń przyjęto $\beta_1 = 2$ oraz $\beta_2 = 1$. Przebiegi wyznaczonej charakterystyki amplitudowej oraz charakterystyki amplitudowej otrzymanej dla tych samych danych wejściowych w wyniku projektowania filtru Nyquista z zastosowaniem aproksymacji równomiernie falistej w pasmie zaporowym [19] są pokazane odpowiednio na rysunkach 6 i 7.



Rys. 6. Charakterystyki amplitudowe filtru Nyquista ($N = 39$) w przypadku: aproksymacji RFSR (linia ciągła) oraz aproksymacji równomiernie falistej w pasmie zaporowym (linia kropkowana)

Jak wynika z tych rysunków, podobnie jak w pierwszym z rozpatrywanych przykładów, w przypadku aproksymacji RFSR można uzyskać mniejsze zafalowania charakterystyki amplitudowej w pasmie przepustowym, a ponadto w znacznej części pasma zaporowego wyraźnie większe tłumienie niż w przypadku aproksymacji równomiernie falistej w zaporowym. Można więc stwierdzić, że aproksymacja RFSR jest szczególnie przydatna do projektowania filtrów Nyquista.

[2] Jest to przykład zaczerpnięty z [14], przy czym w pracy tej zaprezentowano wyniki projektowania filtru Nyquista o równomiernie falistym przebiegu charakterystyki amplitudowej w pasmie zaporowym z zastosowaniem programowania liniowego. W zadaniu projektowania rozpatrywanym w [14] wymagania odnośnie rodzaju przebiegu charakterystyki amplitudowej w pasmie przepustowym nie są sprecyzowane.



Rys. 7. Charakterystyki amplitudowe filtru Nyquista z rys. 6 rozciągnięte w pasmie przepustowym w przypadku: aproksymacji RFSR (linia ciągła) oraz aproksymacji równomiernie falistej w pasmie zaporowym (linia kropkowana)

5. WNIOSKI

W pracy przedstawiono metodę projektowania cyfrowych filtrów SOI o liniowej fazie z zastosowaniem aproksymacji równomiernie falistej w pasmie przepustowym i średniokwadratowej w pasmie zaporowym, przy dodatkowym spełnieniu jednego z następujących warunków: warunku dotyczącego maksymalnej dopuszczalnej amplitudy zafalowań w pasmie przepustowym lub warunku dotyczącego maksymalnej dopuszczalnej amplitudy w pasmie zaporowym. Zaproponowana metoda jest prostsza w implementacji numerycznej niż metoda opisana przez Adamsa [4, 5], gdyż zadanie optymalizacji otrzymane w wyniku jej zastosowania może być rozwiązywane przy użyciu standardowych metod programowania nieliniowego z ograniczeniami, a odpowiednie procedury minimalizacyjne są dostępne w różnych bibliotekach programów. W przypadku metody Adamsa niezbędne jest natomiast opracowanie programu komputerowego realizującego opisany w pracy [4] skomplikowany algorytm wymiany punktów (ang. *multiple exchange algorithm*) oparty na algorytmie Remeza [1]. Ponadto, zaproponowana metoda jest o wiele bardziej elastyczna niż metoda Adamsa, gdyż umożliwia ona uzyskanie różnych rozwiązań kompromisowych pomiędzy aproksymacją równomiernie falistą w pasmie przepustowym i średniokwadratową w pasmie zaporo-

wym. Adams rozpatruje jedynie przypadki takiej aproksymacji równomiernie falistej w pasmie przepustowym, w której dopuszczalna amplituda zafalowań δ_1 w pasmie przepustowym jest taka sama jak otrzymana w przypadku aproksymacji równomiernie falistej w obu pasmach przepustowym i zaporowym. Przy użyciu zaproponowanej metody możliwe jest otrzymanie mniejszej wartości δ_1 w pasmie przepustowym niż w przypadku aproksymacji równomiernie falistej w obu pasmach przepustowym i zaporowym. Odbywa się to jednak kosztem niewielkiego zmniejszenia tłumienia w pasmie zaporowym przy częstotliwościach bliskich f_s oraz niewielkiego poszerzenia pasma przejściowego w porównaniu z przypadkiem aproksymacji równomiernie falistej w obu pasmach. Należy jednak zauważyć, że dla danej długości odpowiedzi impulsowej N filtru istnieje pewna określona minimalna wartość $\delta_1 = \delta_{1\min}$, którą można uzyskać. Otrzymanie mniejszych zafalowań jest możliwe jedynie po zwiększeniu długości odpowiedzi impulsowej N filtru.

W przypadku zaproponowanej metody można również łatwo uwzględnić w zadaniu optymalizacji dwukryterialnej inne dodatkowe ograniczenia. Jako przykład uwzględnienia takich ograniczeń przedstawiono projektowanie filtrów Nyquista. Adams w pracy [4] jedynie wspomina, że w jego metodzie istnieje możliwość uwzględnienia innych dodatkowych warunków. Nie podaje on jednak żadnego przykładu projektowania filtru z uwzględnieniem jakichkolwiek dodatkowych ograniczeń.

Zaproponowana metoda, ze względu na dużą elastyczność i możliwość uwzględnienia różnorodnych przypadków, doskonale nadaje się również do przeprowadzenia badań różnych rodzajów rozwiązań kompromisowych pomiędzy aproksymacją równomiernie falistą i średniokwadratową. Rozwiązania tego rodzaju mogą znaleźć zastosowanie nie tylko przy projektowaniu filtrów cyfrowych o skończonej i nieskończonej odpowiedzi impulsowej, ale również przy projektowaniu anten [5].

6. BIBLIOGRAFIA

1. S. K. Mitra, J. F. Kaiser (ed.): *Handbook for digital signal processing*. New York, John Wiley & Sons, Inc., 1993
2. A. Wojtkiewicz: *Elementy syntezy filtrów cyfrowych*. Warszawa, WNT, 1984
3. S. K. Mitra: *Digital signal processing: a computer-based approach*. McGraw-Hill, 2000
4. J. W. Adams: *FIR digital filters with least-squares stopbands subject to peak-gain constraints*. IEEE Trans. Circuits Syst., 1991, vol. 39, No 4, pp. 376-388
5. J. W. Adams: *Peak-constrained least-squares optimization*. IEEE Trans. Signal Processing, 1998, vol. 46, No. 2, pp. 306-320
6. M. Lang, J. Bamberger: *Nonlinear phase FIR filter design according to the* L_2 *norm with constraints for the complex error*. Signal Processing, 1994, vol. 36, pp. 31-40
7. J. L. Sullivan, J. W. Adams: *PCLS IIR digital filters with simultaneous frequency response magnitude and group delay specifications*. IEEE Trans. Signal Processing, 1998, vol. 46, No 11, pp. 2853-2861
8. W.-S. Lu: *Design of stable IIR digital filters with equiripple passband and peak-constrained least-squares stopbands*. IEEE Trans. Circuits and Syst.- II: Analog and Digital Signal Processing, 1999, vol. 46, No. 11, pp. 1421-1426

9. F. Wysocka-Schillak: *Design of FIR filters with equiripple passband and least-squares stopband.* Proceedings of 10th Mediterranean Electrotechnical Conference, MeleCon 2000, Cyprus, May 29-31, 2000, vol. II – Signal and Image Processing, pp. 607-610
10. F. Wysocka-Schillak: *Projektowanie cyfrowych filtrów SOI o równomiernie falistym przebiegu charakterystyki amplitudowej w pasmie przepustowym przy jednoczesnej minimalizacji energii sygnału w pasmie zaporowym.* Materiały Krajowego Sympozjum Telekomunikacji KST'2000, Bydgoszcz, 6-8.09.2000, ss. 177-186
11. S. Haykin: *Systemy telekomunikacyjne.* Warszawa, WKŁ, 2000
12. R. G. Lyons: *Wprowadzenie do cyfrowego przetwarzania sygnałów.* Warszawa, WKŁ, 1999
13. M. Liu, C. J. Zarowski, F. W. Fairman: *Factorable FIR Nyquist filters with least stopband energy under sidelobe level constraints.* IEEE Trans. Signal Processing, 2000, vol. 48, No. 5, pp. 1495-1498
14. H. Samueli: *On the design of optimal equiripple FIR digital filters for data transmission applications.* IEEE Trans. Circuits and Syst., 1988, vol. CAS-35, No. 12, pp. 1542-1546
15. W. Findeisen, J. Szymanowski, A. Wierzbicki: *Teoria i metody obliczeniowe optymalizacji.* Warszawa, PWN, 1980
16. T. Kręglewski, T. Rogowski, A. Ruszczyński, J. Szymanowski: *Metody optymalizacji w języku FORTRAN.* Warszawa, PWN, 1984
17. W. Press, S. Teukolsky, W. Vatterling, B. Flannery: *Numerical Recipes in Fortran.* Cambridge, Cambridge University Press, 1991
18. *Signal Processing Toolbox User's Guide.* The MathWorks, Inc., 1994
19. F. Wysocka-Schillak: *Design of optimal Nyquist FIR digital filters.* Proc. of the XV-th National Conference „Circuit Theory and Electronic Circuits'', Szczyrk, Oct. 20-23, 1992, vol. 1, pp. 335-340

F. WYSOCKA-SCHILLAK

DESIGN OF LINEAR PHASE FIR FILTERS WITH EQUIRIPPLE PASSBAND AND LEAST-SQUARES STOPBAND

Summary

In the paper, a method for the design of low-pass linear phase FIR filters with the equiripple passband and the least-squares stopband is proposed. In the method, the filter design problem is transformed into an equivalent bicriterion optimization problem. Then the weighted sum strategy is used in order to convert the bicriterion optimization problem into a single criterion one. The design of low-pass linear phase FIR filters is considered in two cases, namely with a constraint on the maximum ripple in the passband and with a constraint on the maximum permissible error in the stopband. The method is also applied to design of Nyquist filters. Four numerical examples demonstrating the usefulness of the proposed approach are given.

Keywords: digital filters, FIR filters, equiripple approximation, least-squares approximation, optimization

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2002, 48, z. 1, ss. 101–115

Estymacja wewnętrznego rozkładu temperatury w tkankach biologicznych za pomocą wieloczęstotliwościowego termografu mikrofalowego

BRONISŁAW STEC, ANDRZEJ DOBROWOLSKI, WALDEMAR SUSEK

Wojskowa Akademia Techniczna, Wydział Elektroniki
ul. Kaliskiego 2, 00–908 Warszawa 49
ADobrowolski@wel.wat.waw.pl

Otrzymano 2001.07.19
Autoryzowano 2001.11.07

W artykule zaprezentowano problematykę związaną z promieniowaniem termicznym ciał rzeczywistych w zakresie mikrofalowym oraz opis własności transmisyjnych tkanek biologicznych rozpatrywanych w aspekcie diagnozowania nowotworu złośliwego sutka. Ponadto przedstawiono algorytm odwrotnej transformacji wieloczęstotliwościowych pomiarów radiometrycznych na rzeczywisty rozkład temperatury wewnątrz badanej tkanki biologicznej oraz zaprezentowano wyniki eksperymentu weryfikującego przedstawioną metodę.

Słowa kluczowe: termograf mikrofalowy, radiometr, promieniowanie termiczne, przestrzenny rozkład temperatury, rak sutka

1. WPROWADZENIE

Termografia mikrofalowa jest gałęzią techniki mikrofalowej związaną z pomiarami i analizą promieniowania termicznego w zakresie częstotliwości mikrofalowych [3]. Zastosowania tej techniki są bardzo rozległe: od detekcji położenia zamaskowanych obiektów wojskowych (np. czołgi, transportery, miny, itp.) poprzez kontrolę temperatury w hipertermii, aż do wykrywania nowotworu złośliwego sutka u kobiet [1].

Promieniowanie termiczne jest promieniowaniem elektromagnetycznym emitowanym przez każde ciało, którego temperatura jest wyższa od zera bezwzględnego. Temperatura ciała ściśle określa moc promieniowania, którego intensywność jest do niej proporcjonalna. Promieniowanie cieplne w paśmie mikrofalowym mierzy się za pomocą radiometru mikrofalowego, który w połączeniu z komputerowym układem przetwarzania i zobrazowania wyników tworzy termograf mikrofalowy.

Medyczne termografy mikrofalowe są urządzeniami, za pomocą których możemy dokonać pomiaru generowanej termicznie przez tkanki podpowierzchniowe emisji mikrofalowej [2]. Ich działanie, w aspekcie diagnozowania nowotworów złośliwych, opiera się na fakcie, że obecność komórek rakowych stymuluje pojawienie się dużej ilości naczyń krwionośnych w ich sąsiedztwie, co w połączeniu z przyspieszonym metabolizmem nowotworu powoduje wzrost temperatury guza i otaczających tkanek.

W odróżnieniu od termografii podczerwonej, która umożliwia pomiar wyłącznie temperatury powierzchni ciała, termografia mikrofalowa pozwala na wykrywanie niejednorodności termicznych o rozmiarach liniowych rzędu milimetrów w głębi próbek biologicznych [9]. Wykrycie tak małych niejednorodności termicznych pozwala na wyeliminowanie interwencji chirurgicznej i zastosowanie bezinwazyjnych metod leczenia.

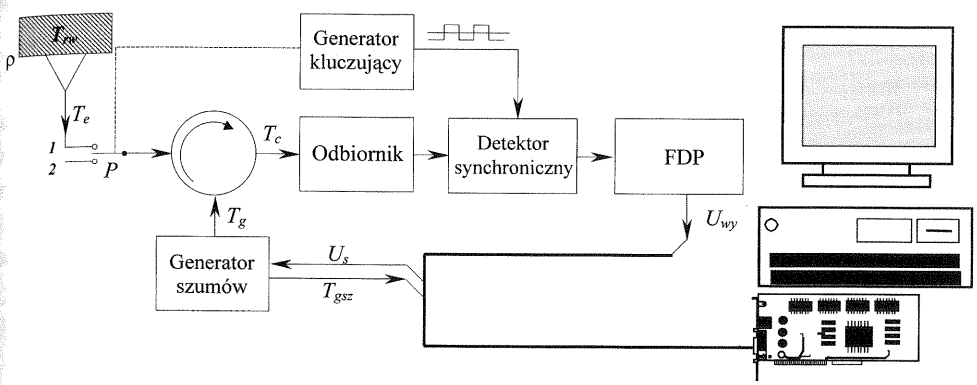
Ze względu na sposób pomiaru mocy szumu termicznego, do którego sprowadza się w zasadzie pomiar temperatury, radiometry mikrofalowe można podzielić na kompensacyjne i modulacyjne (Dicke'a), a w grupie radiometrów modulacyjnych na wyróżnienie zasługują radiometry zrównoważone oraz radiometry z kompensacją współczynnika odbicia.

W radiometrach Dicke'a sygnał wyjściowy zawierający informację o różnicy temperatur jest proporcjonalny do wzmocnienia toru w.cz. radiometru. Z tego powodu wymagana jest duża stałość wzmocnienia i okresowe kalibrowanie radiometru w oparciu o wzorcowe źródło szumów. Istotny jest również fakt, że odczyty w radiometrach Dicke'a zależą nie tylko od skutecznej temperatury szumowej tkanek poddawanych pomiarowi, ale również od niedopasowania pomiędzy anteną a tkanką. Z tego względu radiometry Dicke'a muszą być poddawane rekalibracji praktycznie przed każdym pomiarem. Jest to szczególnie uciążliwe w środowisku klinicznym.

Eliminację błędów pomiaru wynikających z fluktuacji wzmocnienia można uzyskać, stosując radiometr w układzie zrównoważonym, z wewnętrznym sterowanym źródłem szumów (ang. *selfbalanced radiometer*) [9]. Schemat blokowy takiego radiometru, posiadającego ponadto właściwości kompensacji współczynnika odbicia na granicy antena – tkanka, przedstawiono w referacie [8]. W radiometrze tym moc wejściowa porównywana jest z mocą lokalnego generatora szumów. Skutkiem samorównoważenia się urządzenia moc z generatora lokalnego zrównywana jest z mocą wejściową, a napięcie na wyjściu filtru dolnoprzepustowego zmierza do zera. Dzięki temu wyniki pomiarów nie są uzależnione od wzmocnienia radiometru.

W dotychczasowych rozwiązaniach radiometrów konstruowanych na „Wydziale Elektroniki Wojskowej Akademii Technicznej" w Warszawie, sterowanie wewnętrznym generatorem szumów radiometru z kompensacją współczynnika odbicia odbywało się w analogowej pętli regulacji. Sygnał błędu z wyjścia detektora synchronicznego poprzez wzmacniacz pomiarowy i filtr dolnoprzepustowy podawany był na regulacyjne elementy wykonawcze wewnętrznego generatora szumów. Elementami tymi są diody PIN o nieliniowej charakterystyce zmian rezystancji w funkcji płynącego przez nie prądu. Dla zapewnienia quasi liniowej pracy radiometru ograniczano zakres mierzonych temperatur.

Wymienionych ograniczeń i wad nie posiada przedstawiona na rys. 1, sterowana komputerowo cyfrowa pętla regulacji generatora szumów. Napięcie regulacyjne generatora jest zmieniane dopóty dopóki układ regulacji nie osiągnie zerowego napięcia wyjściowego. Nie odgrywa w tym przypadku roli nieliniowość charakterystyk elementów regulacyjnych, gdyż wynik pomiaru temperatury obliczany jest przez program obsługi na podstawie zaimplementowanych charakterystyk regulacyjnych.

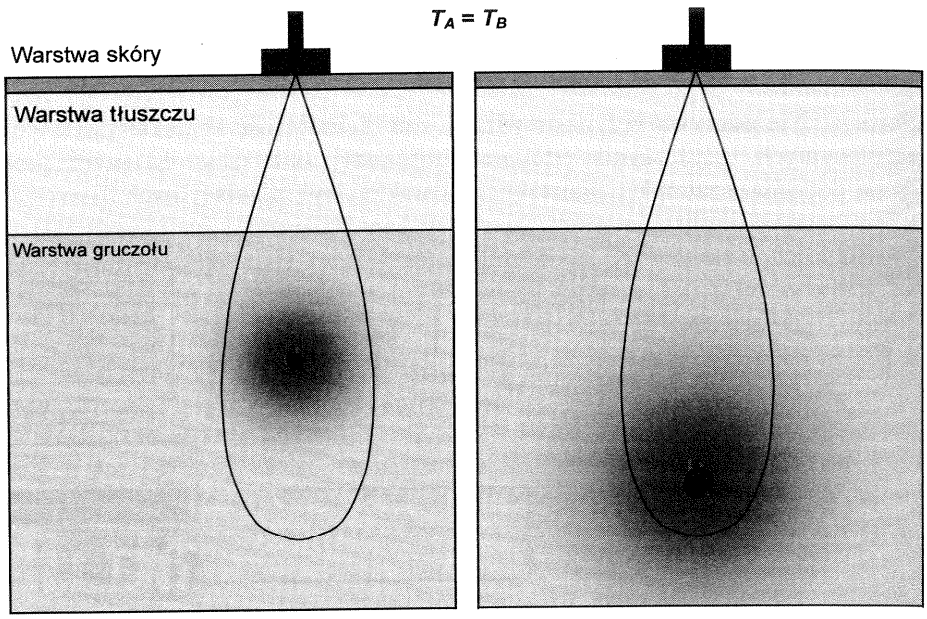


Rys. 1. Schemat blokowy radiometru z kompensacją współczynnika odbicia i cyfrową pętlą regulacji wewnętrznego generatora szumów

Zastosowanie zoptymalizowanego algorytmu sterowania i przetwarzania wyników pośrednich zmniejszyło wrażliwość pomiaru rzeczywistej temperatury badanej tkanki na zmiany czynników zewnętrznych (temperatury otoczenia, parametrów elektrycznych układu, itp.). W omawianym układzie zastosowano odbiornik o bezpośrednim wzmocnieniu, co pozwoliło ograniczyć zakłócenia i poprawić rozdzielczość temperaturową. Dalszą poprawę parametrów użytkowych radiometru zapewnia, realizowana za pomocą procesora sygnałowego, w pełni cyfrowa detekcja synchroniczna i filtracja dolnoprzepustowa.

Monoczęstotliwościowa pasywna radiometria mikrofalowa umożliwia pomiar średniej temperatury na pewnej wybranej głębokości obiektu. Pozwala więc na znalezienie maksimum rozkładu powierzchniowego, co jest równoznaczne z sytuacją, że antena znajduje się bezpośrednio nad źródłem o zwiększonej radiacji. Pomiar na jednej częstotliwości zilustrowany jest poglądowo na rys. 2. Radiometr może wskazywać jednakową skuteczną temperaturę na powierzchni tkanki przy małym ognisku (w sensie rozmiarów i/lub temperatury) położonym płytko pod skórą oraz przy dużym, ale usytuowanym głębiej.

Z punktu widzenia zastosowań niezmiernie interesujące jest zagadnienie estymacji przestrzennego rozkładu temperatury wewnątrz badanego obiektu. Rozwiązanie tak postawionego problemu wykorzystuje radiometrię wieloczęstotliwościową [4÷7]. Podstawą pomiaru jest malejąca w funkcji częstotliwości głębokość wnikania mocy do tkanek biologicznych oraz jednoczesny wzrost intensywności promieniowania termicznego.



Rys. 2. Ilustracja pomiaru monoczęstotliwościowego

Ponieważ rozkład powierzchniowy może być określony poprzez pomiary w kolejnych punktach, w dalszych rozdziałach przestawiona będzie jednowymiarowa analiza wgłębnego profilu rozkładu temperatury, oraz algorytm odwrotnej transformacji wyników wieloczęstotliwościowego pomiaru powierzchniowej jaskrawości temperatury na wgłębny rozkład temperatury.

2. POWIERZCHNIOWA JASKRAWOŚĆ TEMPERATURY ŹRÓDEŁ RZECZYWISTYCH

Jak wykazano w [5÷7] związek łączący rzeczywisty przyrost temperatury źródła ciepła T i odpowiadający mu przyrost jaskrawości temperatury T' na zewnętrznej powierzchni tkanki ma postać

$$T' = T \cdot e(f) \tag{1}$$

przy czym w przypadku próbek biologicznych, emisyjność przestrzenna $e(f)$ zależy od parametrów dielektrycznych obszarów promieniowania, transmisji i odbioru i przy dostatecznie dużym tłumieniu ośrodków wyraża się ogólną zależnością

$$e(f) = \exp\left(-\frac{d_g}{\delta_g}\right) \cdot t_{gt} \cdot \exp\left(-\frac{d_t}{\delta_t}\right) \cdot t_{ts} \cdot \exp\left(-\frac{d_s}{\delta_s}\right) \tag{2}$$

gdzie: d_g, d_t, d_s – długości dróg w warstwach gruczołu, tłuszczu i skóry,
δ_g, δ_t, δ_s – głębokości wnikania w poszczególnych warstwach,
t_{gt}, t_{ts} – współczynniki transmisji mocy na granicach warstw.

Przyrost temperatury mierzony przez radiometr pracujący na częstotliwości f zależy ponadto od współczynnika odbicia mocy na granicy anteny i tkanki ϱ_f i określony jest zależnością

$$T_f = T' \cdot (1 - \varrho_f) \cdot K_T = T' \cdot K_r \tag{3}$$

gdzie K_r oznacza współczynnik uwzględniający dopasowanie anteny do tkanki oraz wypadkowe wzmocnienie radiometru K_T, rozumiane jako stosunek wskazywanego przyrostu temperatury do rzeczywistego przyrostu temperatury szumów wejściowych.

W wyniku kalibracji trzech użytych podczas realizacji eksperymentu prototypów radiometrów pracujących na częstotliwościach $f_1 = 1{,}5$ GHz, $f_2 = 2{,}9$ GHz i $f_3 = 4{,}4$ GHz otrzymano wypadkowe współczynniki przenoszenia systemu o wartościach: $K_{r1} = 1{,}05$, $K_{r2} = 1{,}2$ i $K_{r3} = 0{,}8$.

W praktyce medycznej, a w szczególności w onkologii, nigdy nie ma się do czynienia z punktowym źródłem ciepła, ale z nieznanym rozkładem temperatury $T(z)$. Celem autorów pracy było znalezienie rozkładu $T(z)$ na podstawie szeregu zmierzonych wartości T_{fi}. Jest to typowe zadanie nie mające jednoznacznego rozwiązania. Analiza przedstawionego problemu prowadzi do wniosku, że w celu znalezienia przybliżonego rozwiązania należy przyjąć pewne założenia upraszczające, dotyczące namierzanych źródeł promieniowania termicznego i charakteru estymowanego rozkładu temperatury.

Budową tkanki i specyfiką pierwszych stadiów raka sutka uzasadnione jest założenie, że źródło ciepła ma kształt kulisty, a rozkład temperatury pochodzący od ogniska zanika już w warstwie gruczołu. Powyższe przesłanki wynikające ze statystyk medycznych i potwierdzone sondażowymi pomiarami inwazyjnymi [6, 7] spowodowały wybór krzywej Gaussa jako funkcji opisującej wgłębny rozkład temperatury. Wybrany rozkład przedstawiony jest na rys. 3, a przyjęty zapis analityczny ma postać

$$T(z) = T_{z,r} \cdot \exp\left[-\left(\frac{z - d_g}{\sigma}\right)^2\right] \tag{4}$$

Ponieważ przyrost rozkładu temperatury pochodzący od ogniska zanika do zera już w warstwie gruczołu, zależność (3) uzyskuje postać

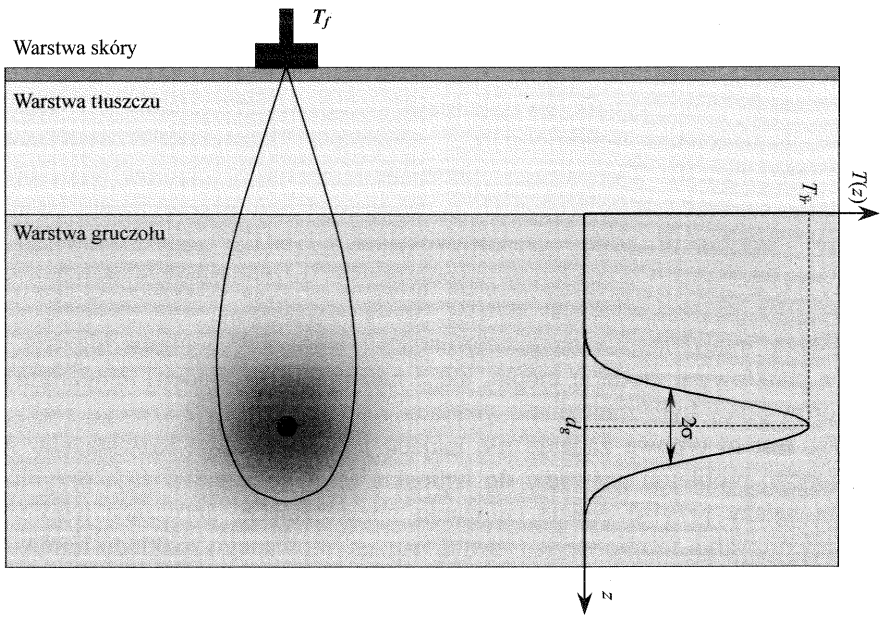
$$T_f = T_g \cdot t_{gt} \cdot \exp\left(-\frac{d_t}{\delta_t}\right) \cdot t_{ts} \cdot \exp\left(-\frac{d_s}{\delta_s}\right) \cdot K_r \tag{5}$$

gdzie jaskrawość temperatury T_g na granicy warstw gruczołu i tłuszczu określona jest całką z rozkładu temperatury $T(z)$ w granicach $\langle 0, \infty \rangle$ z uwzględnieniem współczynnika przenoszenia

$$\xi(z) = \exp\left(-\frac{z}{\delta_g}\right) \tag{6}$$

Korzystając z (4) i (6) otrzymujemy

$$T(\xi) = T_{zr} \cdot e^{-\left(\frac{d_g}{\sigma}\right)^2} \cdot \xi^{-\frac{\delta_g}{\sigma^2}(\delta_g \ln\xi + 2d_g)} \tag{7}$$



Rys. 3. Rozkład temperatury wewnątrz tkanki

Całkowanie względem zmiennej z od 0 do ∞ jest równoważne całkowaniu po ξ w granicach $\langle 0, 1\rangle$ i w konsekwencji jaskrawość temperatury na granicy gruczołu i tłuszczu określona jest zależnością

$$T_g = \int_0^1 T(\xi)\, d\xi = T_{zr} \cdot e^{-\frac{d_g^2}{\sigma^2}} \cdot \int_0^1 \xi^{-\frac{\delta_g}{\sigma^2}(\delta_g \ln\xi + 2d_g)}\, d\xi \tag{8}$$

Nie istnieje niestety wyrażenie opisujące funkcję pierwotną funkcji podcałkowej z powyższej zależności, ale można wyprowadzić zależność na poszukiwaną całkę oznaczoną

$$T_g = \frac{T_{zr}\sigma\sqrt{\pi}}{2\delta_g} \exp\left(\frac{\sigma^2 - 4\delta_g d_g}{4\delta_g^2}\right) \cdot \left[\operatorname{erf}\left(\frac{2\delta_g d_g - \sigma^2}{2\sigma\delta_g}\right) + 1\right] \tag{9}$$

która dla

$$d_g \geq d_{g\min} = 2\sigma + \frac{\sigma^2}{2\delta_g} \tag{10}$$

sprowadza się do postaci

$$T_g = \frac{T_{źr}\sigma\sqrt{\pi}}{\delta_g}\exp\left(\frac{\sigma^2 - 4\delta_g d_g}{4\delta_g^2}\right) \tag{11}$$

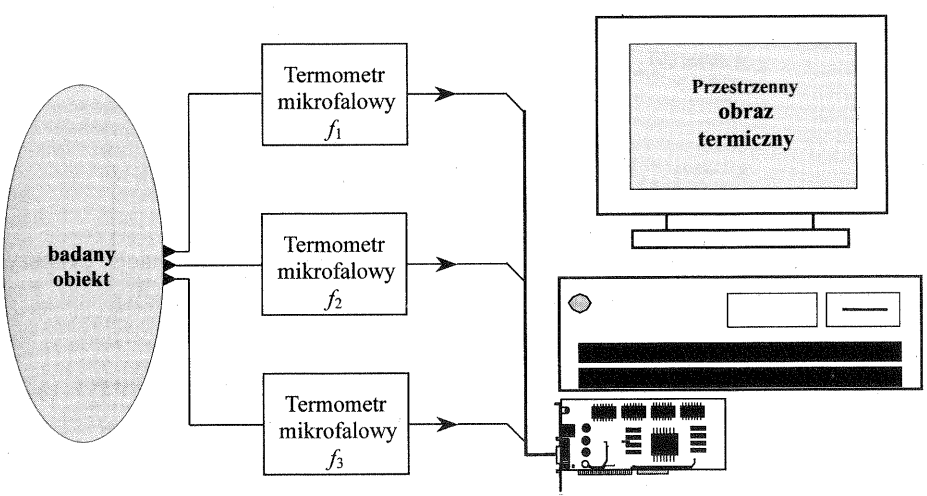
Korzystając z (5) i (11), otrzymujemy zależność wiążącą zmianę powierzchniowej jaskrawości temperatury mierzoną radiometrem pracującym na częstotliwości f ze zmianą rzeczywistego rozkładu temperatury

$$T_f = \frac{T_{źr}\sigma\sqrt{\pi}}{\delta_g}\exp\left(\frac{\sigma^2 - 4\delta_g d_g}{4\delta_g^2}\right)\cdot t_{gt}\cdot\exp\left(-\frac{d_t}{\delta_t}\right)\cdot t_{ts}\cdot\exp\left(-\frac{d_s}{\delta_s}\right)\cdot K_r \tag{12}$$

3. TRANSFORMACJA ODWROTNA

3.1. JEDNORODNA TKANKA MIĘŚNIOWA

W przypadku pojedynczej warstwy mięśniowej zależność (12) opisująca powierzchniową jaskrawość temperatury mierzoną radiometrem pracującym na częstotliwości f_i w funkcji wgłębnego rozkładu temperatury przyjmuje postać



Rys. 4. Trójczęstotliwościowy termograf mikrofalowy

$$T_{fi} = \frac{T_{zr}\sigma\sqrt{\pi}}{\delta_{mi}} \exp\left(\frac{\sigma^2 - 4\delta_{mi} d_m}{4\delta_{mi}^2}\right) \cdot K_{ri} \tag{13}$$

Znając, wyznaczone w procesie kalibracji współczynniki K_{ri} dla trzech radiometrów pracujących na różnych częstotliwościach i odpowiadające im głębokości wnikania δ_{mi}, można wyznaczyć rzeczywisty rozkład temperatury realizując pomiar wieloczęstotliwościowy (rys. 4).

Dokonując pomiaru wieloczęstotliwościowego z wykorzystaniem trzech radiometrów i korzystając z (13), otrzymujemy układ równań nieliniowych o postaci

$$\begin{cases} T_{f1} = \dfrac{T_{zr}\sigma\sqrt{\pi}}{\delta_{m1}} \exp\left(\dfrac{\sigma^2 - 4\delta_{m1} d_m}{4\delta_{m1}^2}\right) \cdot K_{r1} \\ T_{f2} = \dfrac{T_{zr}\sigma\sqrt{\pi}}{\delta_{m2}} \exp\left(\dfrac{\sigma^2 - 4\delta_{m2} d_m}{4\delta_{m2}^2}\right) \cdot K_{r2} \\ T_{f3} = \dfrac{T_{zr}\sigma\sqrt{\pi}}{\delta_{m3}} \exp\left(\dfrac{\sigma^2 - 4\delta_{m3} d_m}{4\delta_{m3}^2}\right) \cdot K_{r3} \end{cases} \tag{14}$$

Rozwiązaniem przedstawionego układu są wyrażenia określające rzeczywisty rozkład temperatury w badanej tkance:

$$d_m = \frac{\ln\left(\dfrac{T_{f1}\cdot K_{r2}\cdot \delta_{m1}}{T_{f2}\cdot K_{r1}\cdot \delta_{m2}}\right)\cdot \delta_{m2}^2(\delta_{m2}^2 - \delta_{m3}^2) - \ln\left(\dfrac{T_{f2}\cdot K_{r3}\cdot \delta_{m2}}{T_{f3}\cdot K_{r2}\cdot \delta_{m3}}\right)\cdot \delta_{m3}^2(\delta_{m1}^2 - \delta_{m2}^2)}{\delta_{m1}^2(\delta_{m2} - \delta_{m3}) - \delta_{m2}^2(\delta_{m1} - \delta_{m3}) + \delta_{m3}^2(\delta_{m1} - \delta_{m2})}$$

$$\sigma = 2\delta_{m1}\delta_{m2}\sqrt{\frac{d_m(\delta_{m2}^{-1} - \delta_{m1}^{-1}) - \ln\left(\dfrac{T_{f1}\cdot K_{r2}\cdot \delta_{m1}}{T_{f2}\cdot K_{r1}\cdot \delta_{m2}}\right)}{\delta_{m1}^2 - \delta_{m2}^2}} \tag{15}$$

$$T_{zr} = \frac{T_{f1}\delta_{m1}}{\sigma K_{r1}\sqrt{\pi}} \exp\left(\frac{4\delta_{m1} d_m - \sigma^2}{4\delta_{m1}^2}\right)$$

Przedstawione zależności zostały wykorzystane podczas realizacji eksperymentu weryfikującego zaprezentowaną analizę.

3.2. TKANKA ZŁOŻONA Z WARSTW GRUCZOŁU, TŁUSZCZU I SKÓRY

W przypadku, gdy badana tkanka (sutek) złożona jest z warstw gruczołu, tłuszczu i skóry, w celu estymacji wewnętrznego rozkładu temperatury, należy dokonać pomiaru w systemie czteroczęstotliwościowym.

W rozważanym przypadku zależność (12) opisująca powierzchniową jaskrawość temperatury mierzoną radiometrem pracującym na częstotliwości f_i w funkcji wgłębnego rozkładu temperatury przyjmuje postać

$$T_{fi} = \frac{T_{zr}\sigma\sqrt{\pi}}{\delta_{gi}} \exp\left(\frac{\sigma^2 - 4\delta_{gi} d_g}{4\delta_{gi}^2}\right) \cdot t_{gti} \cdot \exp\left(-\frac{d_t}{\delta_{ti}}\right) \cdot t_{tsi} \cdot \exp\left(-\frac{d_s}{\delta_{si}}\right) \cdot K_{ri} \tag{16}$$

Ponieważ warstwę skóry w badanym miejscu – ze względu na jej znikomą grubość – można potraktować jako warstwę cienką, zależność (16) upraszcza się do postaci

$$T_{fi} = \frac{T_{zr}\sigma\sqrt{\pi}}{\delta_{gi}} \exp\left(\frac{\sigma^2 - 4\delta_{gi} d_g}{4\delta_{gi}^2}\right) \cdot t_{gti} \cdot \exp\left(-\frac{d_t}{\delta_{ti}}\right) \cdot K_{ri} \tag{17}$$

W rozpatrywanym paśmie częstotliwości (1,5 GHz÷4,4 GHz) przy kącie padania z zakresu ±25° współczynnik transmisji mocy na granicy gruczołu i tłuszczu $t_{gti} \approx 0{,}96$. W konsekwencji zależność (17) przyjmuje postać

$$T_{fi} = \frac{T_{zr}\sigma\sqrt{\pi}}{\delta_{gi}} \exp\left(\frac{\sigma^2 - 4\delta_{gi} d_g}{4\delta_{gi}^2}\right) \cdot \exp\left(-\frac{d_t}{\delta_{ti}}\right) \cdot 0{,}96\, K_{ri} \tag{18}$$

Dokonując pomiaru wieloczęstotliwościowego z wykorzystaniem czterech radiometrów i korzystając z (18), otrzymujemy układ równań nieliniowych o postaci

$$\begin{cases} T_{f1} = \dfrac{T_{zr}\sigma\sqrt{\pi}}{\delta_{g1}} \exp\left(\dfrac{\sigma^2 - 4\delta_{g1} d_g}{4\delta_{g1}^2}\right) \cdot \exp\left(-\dfrac{d_t}{\delta_{t1}}\right) \cdot 0{,}96\, K_{r1} \\ T_{f2} = \dfrac{T_{zr}\sigma\sqrt{\pi}}{\delta_{g2}} \exp\left(\dfrac{\sigma^2 - 4\delta_{g2} d_g}{4\delta_{g2}^2}\right) \cdot \exp\left(-\dfrac{d_t}{\delta_{t2}}\right) \cdot 0{,}96\, K_{r2} \\ T_{f3} = \dfrac{T_{zr}\sigma\sqrt{\pi}}{\delta_{g3}} \exp\left(\dfrac{\sigma^2 - 4\delta_{g3} d_g}{4\delta_{g3}^2}\right) \cdot \exp\left(-\dfrac{d_t}{\delta_{t3}}\right) \cdot 0{,}96\, K_{r3} \\ T_{f4} = \dfrac{T_{zr}\sigma\sqrt{\pi}}{\delta_{g4}} \exp\left(\dfrac{\sigma^2 - 4\delta_{g4} d_g}{4\delta_{g4}^2}\right) \cdot \exp\left(-\dfrac{d_t}{\delta_{t4}}\right) \cdot 0{,}96\, K_{r4} \end{cases} \tag{19}$$

Numeryczne rozwiązanie przedstawionego układu pozwala na wyznaczenie wszystkich parametrów poszukiwanego rozkładu temperatury w badanej tkance (T_{zr}, σ, d_g, d_t). Rzeczywista głębokość źródła ciepła określona jest sumą $d = d_g + d_t + d_s \approx d_g + d_t$.

Ze względu na łatwość oszacowania grubości warstwy tłuszczu oraz wykorzystując założenie, że rozkład temperatury pochodzący od ogniska zanika już w warstwie

gruczołu, można wyznaczyć poszukiwany rozkład za pomocą pomiaru trójczęstotliwościowego, wprowadzając do zależności (17) wypadkowy współczynnik określony wzorem

$$K_{wi} = t_{gti} \cdot \exp\left(-\frac{d_t}{\delta_{ti}}\right) \cdot K_{ri} \tag{20}$$

Wówczas

$$T_{fi} = \frac{T_{źr}\sigma\sqrt{\pi}}{\delta_{gi}} \exp\left(\frac{\sigma^2 - 4\delta_{gi} d_g}{4\delta_{gi}^2}\right) \cdot K_{wi} \tag{21}$$

W wyniku pomiaru na trzech częstotliwościach, korzystając z (21), otrzymujemy

$$\begin{cases} T_{f1} = \dfrac{T_{źr}\sigma\sqrt{\pi}}{\delta_{g1}} \exp\left(\dfrac{\sigma^2 - 4\delta_{g1} d_g}{4\delta_{g1}^2}\right) \cdot K_{w1} \\ T_{f2} = \dfrac{T_{źr}\sigma\sqrt{\pi}}{\delta_{g2}} \exp\left(\dfrac{\sigma^2 - 4\delta_{g2} d_g}{4\delta_{g2}^2}\right) \cdot K_{w2} \\ T_{f3} = \dfrac{T_{źr}\sigma\sqrt{\pi}}{\delta_{g3}} \exp\left(\dfrac{\sigma^2 - 4\delta_{g3} d_g}{4\delta_{g3}^2}\right) \cdot K_{w3} \end{cases} \tag{22}$$

Rozwiązaniem przedstawionego układu są wyrażenia określające rzeczywisty rozkład temperatury w badanej tkance:

$$d_g = \frac{\ln\left(\dfrac{T_{f1} \cdot K_{w2} \cdot \delta_{g1}}{T_{f2} \cdot K_{w1} \cdot \delta_{g2}}\right) \cdot \delta_{g2}^2 (\delta_{g2}^2 - \delta_{g3}^2) - \ln\left(\dfrac{T_{f2} \cdot K_{w3} \cdot \delta_{g2}}{T_{f3} \cdot K_{w2} \cdot \delta_{g3}}\right) \cdot \delta_{g3}^2 (\delta_{g1}^2 - \delta_{g2}^2)}{\delta_{g1}^2 (\delta_{g2} - \delta_{g3}) - \delta_{g2}^2 (\delta_{g1} - \delta_{g3}) + \delta_{g3}^2 (\delta_{g1} - \delta_{g2})}$$

$$\sigma = 2\delta_{g1}\delta_{g2} \sqrt{\frac{d_g (\delta_{g2}^{-1} - \delta_{g1}^{-1}) - \ln\left(\dfrac{T_{f1} \cdot K_{w2} \cdot \delta_{g1}}{T_{f2} \cdot K_{w1} \cdot \delta_{g2}}\right)}{\delta_{g1}^2 - \delta_{g2}^2}} \tag{23}$$

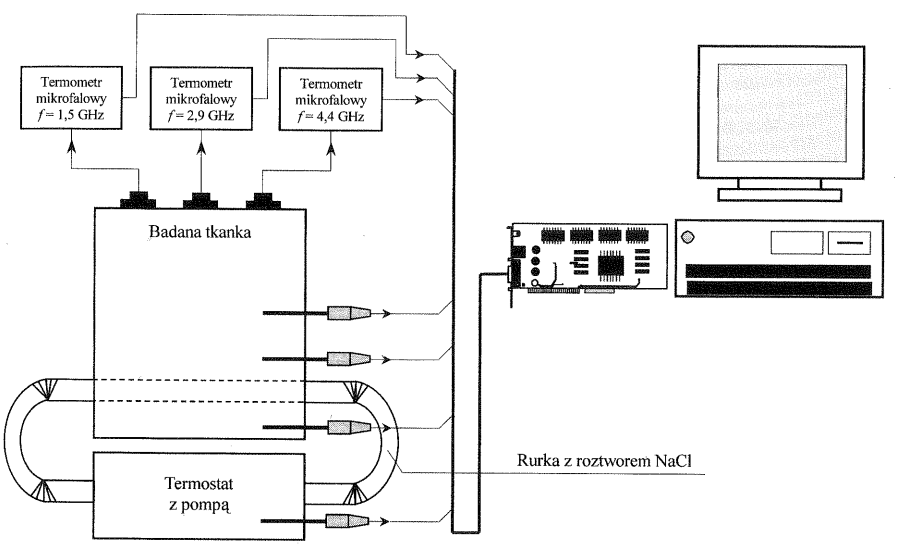
$$T_{źw} = \frac{T_{f1}\delta_{g1}}{\sigma K_{w1}\sqrt{\pi}} \exp\left(\frac{4\delta_{g1} d_g - \sigma^2}{4\delta_{g1}^2}\right)$$

Analiza numeryczna przedstawionych w niniejszym rozdziale zależności prowadzi do wniosku, że wyniki oszacowania nieznanego rozkładu temperatury silnie zależą od dokładności pomiaru T_{fi}. Jest to efekt występujących w końcowych wyrażeniach

szybkozmiennych, w zależności od argumentu, funkcji eksponencjalnych. Wskazane zatem wydaje się dokonanie pomiaru na dodatkowych częstotliwościach w celu sprecyzowania wyników obliczeń. Wiąże się to dodatkowo z doborem optymalnego algorytmu przybliżonego rozwiązywania nadmiarowego układu równań nieliniowych.

4. OPIS STANOWISKA POMIAROWEGO

Podczas realizacji eksperymentu w charakterze badanej tkanki wykorzystano szynkę wieprzową, a jako źródło ciepła zastosowano rurkę polipropylenową o średnicy 5 mm z przepływającym wewnątrz 1,5% wodnym roztworem NaCl o temperaturze regulowanej za pomocą termostatu. W zakresie temperatur pomiarowych (20°C÷55°C) konduktywność roztworu wynosi około 2 S/m, a stała dielektryczna mieści się w granicach od 80 do 68. Takie parametry zapewniają bardzo dobry współczynnik transmisji na granicy roztworu z tkanką, wynoszący w pomiarowym zakresie częstotliwości około 0,99. Wpływ rurki na współczynnik transmisji można pominąć ze względu na efekt tunelowania (grubość ścianki wynosi 0,1 mm).

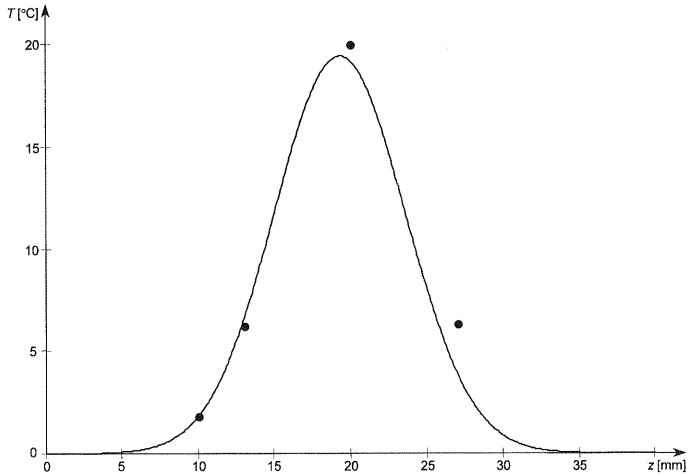


Rys. 5. Stanowisko pomiarowe

Stanowisko pomiarowe przedstawiono poglądowo na rys. 5. Do kontroli rozkładu temperatury wewnątrz tkanki wykorzystano platynowe sondy podskórne typu *RTD* (ang. *Resistor Temperature Device*). Pomiarów dokonywano za pomocą trzech radiometrów pracujących na częstotliwościach f_1 = 1,5 GHz, f_2 = 2,9 GHz i f_3 = 4,4 GHz.

Zastosowaną tkankę przebadano pod kątem własności propagacyjnych i określono średnie głębokości wnikania mocy dla trzech wykorzystywanych w trakcie realizacji eksperymentu radiometrów: $\delta_1 = 6{,}6$ mm, $\delta_2 = 3{,}8$ mm i $\delta_3 = 3$ mm.

Wyniki pomiarów radiometrycznych i bezpośrednich zaprezentowane są na rys. 7. Linią ciągłą zobrazowano wyliczony na podstawie wskazań radiometrów rozkład temperatury, zaś za pomocą punktów przedstawiono wyniki pomiarów bezpośrednich dokonanych sondami *RTD*.



Rys. 6. Wyniki pomiarów i obliczeń wewnętrznego rozkładu temperatury

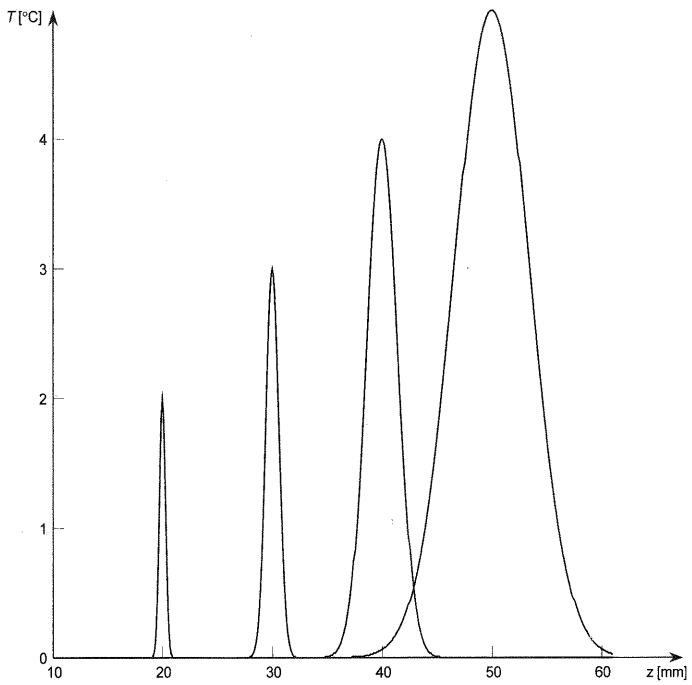
Uzyskane wyniki w pełni potwierdzają poprawność przedstawionej w niniejszym artykule metody uzyskiwania wgłębnego profilu rozkładu temperatury na bazie wielo-częstotliwościowych pomiarów radiometrycznych.

5. PODSUMOWANIE

Celem artykułu było przedstawienie metody pomiarowej pozwalającej na skonstruowanie przestrzennego termografu mikrofalowego. Analiza teoretyczna oraz eksperymenty potwierdziły wstępne prognozy, które stanowiły podstawę podjęcia próby opracowania metody wyznaczania przestrzennego rozkładu temperatury wewnątrz tkanek biologicznych. W przypadku jednorodnej warstwy mięśniowej można za pomocą pomiaru na trzech różnych częstotliwościach wyznaczyć rzeczywisty wgłębny profil temperatury (punkt 3.1). Dla trzech warstw: gruczołu, tłuszczu, i skóry konieczny jest pomiar wieloczęstotliwościowy za pomocą radiometrów pracujących na czterech różnych częstotliwościach (punkt 3.2), przy czym zastosowanie odpowiednich zabiegów dodatkowych umożliwia estymację rozkładu temperatury z wykorzystaniem jedynie trzech radiometrów.

Szacunki wykonane podczas realizacji pracy wskazują, że w przypadku badań onkologicznych (estymacja położenia i rozmiarów raka sutka) zastosowany wielopasmowy system radiometryczny zdolny jest do oceny guzów nowotworowych o rozmiarach i lokalizacji zilustrowanych na rys. 7. Im mniejszy – co do temperatury i rozmiarów – jest nowotwór tym płycej musi być ulokowany, gdyż powierzchniowa jaskrawość temperatury na powierzchni tkanki musi być większa od czułości wszystkich radiometrów wchodzących w skład systemu pomiarowego.

Należy zaznaczyć, że o granicznych możliwościach całego systemu estymacji położenia i rozmiarów źródeł ciepła decyduje czułość radiometru pracującego na najwyższej częstotliwości. Wynika to z faktu, że głębokość wnikania mocy w głąb tkanki biologicznej jest dla tej częstotliwości najmniejsza. Z drugiej strony radiometr pracujący na najmniejszej częstotliwości będzie determinował warunki detekcji, czyli możliwość stwierdzenia obecności źródła ciepła w polu widzenia anteny.



Rys. 7. Graniczne możliwości badanego termografu (szacunki teoretyczne z warstwą tłuszczu o szerokości 10 mm)

Z punktu widzenia błędu pomiarowego wskazana jest pewna nadmiarowość informacji, która uodparnia cały system na błędy w interpretacji wyników. Na podstawie dotychczasowych badań autorzy uważają, że uzasadnione jest wprowadzenie w tym celu piątej częstotliwości leżącej w okolicy 1 GHz (największa głębokość wnikania).

6. BIBLIOGRAFIA

1. B. Bocquet, J. C. Van De Velde, A. Mamouni, Y. Leroy, G. Giaux, J. Delannoy, D. Delvalee: *Microwave radiometric imaging at 3GHz for the exploration of breast tumours.* IEEE Transactions On MTT, 38(6), 1990, pp. 791÷793
2. Y. Leroy, A. Mamouni, J. C. Van De Velde, B. Bocquet, G. Giaux, J. Dellanoy: *Non – invasive measurement of subcutaneous tissue temperature by microwave radiometry.* Innov. Tech. Biol. Med., 12, 1991, pp. 155÷162
3. A. Mamouni, Y. Leroy, B. Bocquet, J. C. Van De Velde, F. Gelin: *Computation of near – field microwave radiometric signals: definition and experimental verification.* IEEE Transactions On MTT, 39(1), 1991, pp. 124÷132
4. S. Mizushina, K. Abe, H. Ohba, K. Kondoh, Y. Nakamura, T. Sugiura: *Multifrequency microwave radiometry for medical applications.* Res. Inst. of Electron., Shizuoka Univ., Hamamatsu, Japan, 1996, 4, pp. 1257÷1260
5. B. Stec, A. Dobrowolski: *Wieloczęstotliwościowa termografia mikrofalowa.* Kwartalnik Elektroniki i Telekomunikacji PAN, 45(2), Warszawa 1999, pp. 225÷233
6. B. Stec, A. Dobrowolski: *Analiza i pomiary wewnętrznego rozkładu temperatury tkanek biologicznych z wykorzystaniem wieloczęstotliwościowego termografu mikrofalowego.* Biuletyn WAT, Rok XLIX, 6 (574), 2000, pp. 5÷18
7. B. Stec, A. Dobrowolski: *Estimation of internal distribution of temperature inside biological tissues by means of multifrequency microwave thermograph.* 13th International Conference on Microwaves, Radar and Wireless Communications (MIKON'2000), Wrocław 2000, pp. 577÷580
8. B. Stec, W. Susek: *A 4.4 GHz microwave thermometer with compensation of reflection coefficient.* 13th International Conference on Microwaves, Radar and Wireless Communications (MIKON'2000), Wrocław 2000, pp. 453÷456
9. F. Sterzer: *Microwave radiometers for non – invasive measurements of subsurface tissue temperatures.* Automedica, 8(4), 1987, pp. 203÷211

B. STEC, A. DOBROWOLSKI, W. SUSEK

ESTIMATION OF INTERNAL DISTRIBUTION OF TEMPERATURE INSIDE BIOLOGICAL TISSUES BY MEANS OF MULTIFREQUENCY MICROWAVE THERMOGRAPH

Summary

The passive microwave thermography is based on measurement of thermal radiation emitted by each body, which has the temperature higher than the absolute zero. The greatest intensity of radiation is in the infrared, but high attenuation of tissue in this range limits application of the infrared thermography only to measurements of skin temperature. In the microwaves the intensity of radiation is about ten million times less but attenuation of tissue is low.

The monofrequency radiometry enables measurement of average temperature of a certain area. Therefore, we do not know whether the heat source is cool and not deep under the skin or perhaps it is hot but deeply situated. In both cases the temperature brightness on the external surface may be equal. We only know that there is an area of increased temperature under the antenna, which may indicate the presence of tumour.

From the practical point of view the problem of estimation of spatial temperature distribution inside the investigated object is particularly interesting. The presented solution uses the power thermography on different frequencies. This method is based on the increasing intensity of thermal radiation and at the same time on the decreasing depth of penetration into biological tissues vs. frequency.

The correctness of the presented analysis has been confirmed by the described experiment. The obtained results indicate a possibility of non-invasive detecting and measuring of spatial temperature distribution inside a human body by means of multifrequency microwave thermograph. We hope that the presented method will be used in oncology and other fields of medicine.

Keywords: microwave thermograph, radiometer, thermal radiation, spatial temperature distribution, breast cancer

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2002, 48, z. 1, ss. 117–160

Wprowadzenie do zagadnień translacji automatycznej

MIROSŁAW GAJER

Katedra Automatyki AGH
Al. Mickiewicza 30, 30-059 Kraków
e-mail:mgajer@ia.agh.edu.pl

Otrzymano 2001.07.27
Autoryzowano 2001.10.16

Opracowanie programów komputerowych, które byłyby w stanie dokonywać automatycznych przekładów pomiędzy naturalnymi językami było od wielu lat niedościgłym celem wielu ośrodków badawczych. Również pomyślne rozwiązanie tego problemu posiadałoby poważne skutki natury politycznej, socjologicznej i naukowej. Translacja automatyczna była również jedną z najwcześniejszych aplikacji, jakie zostały zasugerowane dla komputerów. Jednakże obrócenie marzeń o translacji automatycznej w rzeczywistość okazało się niezwykle trudnym i znacznie ciekawszym, niż uprzednio sądzono, zadaniem. Niemniej jednak obecnie dostępnych jest już wiele programów komercyjnych, które z powodzeniem realizują stawiane przed nimi zadania. W artykule przedstawiono podstawy translacji automatycznej. Naświetlono historię rozwoju takich systemów i podano argumenty na rzecz ich praktycznej przydatności. Szczególną uwagę zwrócono na te zagadnienia, których rozwiązanie sprawia w przypadku systemów translacji automatycznej najwięcej trudności. W artykule zamieszczono także opis działania nowoczesnych systemów translacji automatycznej i wskazano na kierunki badań, które wydają się być najbardziej obiecujące. Autor wyraża nadzieję, że artykuł niniejszy dobrze spełni role przewodnika wprowadzającego czytelnika w świat systemów translacji automatycznej, wypełniając jednocześnie panującą w tej dziedzinie lukę w piśmiennictwie polskim.

Słowa kluczowe: przetwarzanie języka naturalnego, lingwistyka komputerowa, translacja automatyczna

1. WPROWADZENIE

W ostatnich latach w obszarze praktycznych zastosowań informatyki zaobserwować można było spore zainteresowanie technikami przetwarzania języka naturalnego. W licznych ośrodkach naukowych na świecie prowadzone były badania, których celem było opracowanie technik przetwarzania języka naturalnego przez komputer [32]. Badania w tym zakresie prowadzone są w różnych kierunkach. Wyróżnić można w tym miejscu,

między innymi, prace zmierzające do konstrukcji systemów rozpoznawania sygnału mowy i jego konwersji do postaci tekstowej [1]. Ponadto prowadzone są badania nad opracowaniem systemów, które pozwolą na interpretację poleceń wydawanych w języku naturalnym, dzięki czemu systemy komputerowe mogą zyskać nowy typ interfejsu użytkownika [2]. Inną grupę urządzeń stanowią tzw. automatyczne systemy dialogowe, które umożliwiają użytkownikowi zadawanie pytań w języku naturalnym. Systemy takie interpretują zapytania użytkownika i w miarę swoich możliwości (dość zresztą ograniczonych) starają się udzielić odpowiedzi na zadane pytanie [3]. Na świecie działa już wiele takich systemów, które pozwalają na zadawanie w języku naturalnym zapytań odnośnie prognozy pogody na najbliższy dzień, bieżących cen akcji na giełdach papierów wartościowych i kursów walut, a także dokonywanie rezerwacji miejsc w hotelach i biletów lotniczych [4].

Ponieważ każdy język naturalny jest systemem niezwykle bogatym, charakteryzującym się wielkim stopniem złożoności, jego maszynowa interpretacja nastręcza wiele problemów. Zwłaszcza dużym utrudnieniem jest wieloznaczność wypowiedzi (ang. *ambiguity*), będąca immanentną cechą każdego języka naturalnego, która polega na tym, że ta sama wypowiedź może być interpretowana na wiele różnych sposobów, w zależności od tego, jakie znaczenia słów zostaną wybrane spośród oferowanych przez słownik możliwości i w jaki sposób zostanie przeprowadzona analiza gramatyczna i składniowa wypowiedzi [5]. W ogólności są to zagadnienia niezwykle trudne, ponieważ często do rozwikłania sensu wypowiedzi nie wystarczy sama wiedza o języku (ang. *linguistic knowledge*), ale także potrzebna jest ogólna wiedza o świecie (ang. *real-world knowledge*) oraz poczucie zdrowego rozsądku (ang. *common sence*), w które trudno jest wyposażyć jakikolwiek komputer [6]. Z wymienionych powyżej powodów często w komunikacji człowieka z maszyną stosowane są tzw. kontrolowane języki naturalne, stanowiące pewien ograniczony podzbiór języków naturalnych, z których się pierwotnie wywodzą [7]. W językach kontrolowanych poprzez nałożenie sztucznych ograniczeń na gramatykę, gdzie pewne poprawne gramatycznie konstrukcje języka naturalnego nie są dopuszczalne (ze względu na zbyt wielki stopień ich komplikacji), a także ograniczenie leksyki do pewnej wybranej dziedziny (każde ze słów posiada tylko jedno znaczenie w ramach obranej dziedziny, której język kontrolowany dotyczy), problem wieloznaczności wypowiedzi zostaje w dużej mierze wyeliminowany, ale niestety i tak nie można się go pozbyć do końca [8].

Innym obszarem zastosowań technik przetwarzania języka naturalnego jest translacja automatyczna, historycznie zwana również translacją maszynową MT (ang. *Machine Translation*). Translacja maszynowa jest nauką (a właściwie sztuką) programowania komputerów w taki sposób, aby mogły dokonywać przekładu tekstu zapisanego w pewnym języku naturalnym (np. koreańskim) na wybrany inny język naturalny (np. duński) [9]. Automatyczna translacja zaliczana jest do dziedziny systemów sztucznej inteligencji AI (ang. *Artificial Intelligence*), których celem jest symulowanie, przy pomocy maszyn cyfrowych, intelektualnych możliwości człowieka. Translacja maszynowa jest dziedziną wiedzy o ponad półwiekowej tradycji, która po pierwszych dość spektakularnych sukcesach przeżywała jeszcze liczne wzloty i upadki [10]. Obecnie istnieje już wiele

komercyjnie dostępnych systemów translacji maszynowej oraz prowadzone są badania podstawowe w wielu różnych ośrodkach na świecie, z których na czoło wysuwają się instytucje japońskie, gdzie ze względu na znaczenie jakie ma potężna bariera językowa, istniejąca pomiędzy Japonią (znajomość języka angielskiego jest w społeczeństwie japońskim bardzo słabo rozpowszechniona) a innymi wysoko rozwiniętymi krajami świata. Z tego powodu w Japonii badania nad translacją automatyczną posiadają bardzo wysoki priorytet oraz uzyskują silne wsparcie finansowe instytucji rządowych [11]. Badania nad translacją maszynową prowadzone są także w Stanach Zjednoczonych i w Europie, ale i tak co bardziej złośliwi twierdzą, że translacja maszynowa jest to coś, co robi się w Japonii, coś o czym w Stanach Zjednoczonych się pisze i coś o czym w Europie można tylko pomarzyć [12].

Jeżeli chodzi o Polskę, to trzeba niestety przyznać, że translacja automatyczna należy raczej do zaniedbanych obszarów nauki. Prace badawcze prowadzone są w nielicznych ośrodkach (wymienić w tym miejscu należy projekt realizowany na Uniwersytecie Śląskim, którego celem jest stworzenie systemu tłumaczącego w sposób automatyczny akty prawne Unii Europejskiej z języka angielskiego na polski [13]), ukazuje się bardzo niewiele publikacji na ten temat [14], nie ma żadnej specjalistycznej konferencji naukowej poświeconej tym zagadnieniom oraz nie ma żadnego podręcznika lub monografii naukowej wprowadzającej w zagadnienia translacji maszynowej. Jeżeli chodzi o komercyjnie dostępne programy, dokonujące automatycznej translacji to oferowanych jest (nawet bezpłatnie w sieci Internet [15]) kilka propozycji, ale dotyczą one wyłącznie przekładów pomiędzy językiem polskim i angielski (w obie strony), rzadziej pomiędzy językiem polskim i niemieckim. Niestety jakość tych programów pozostawia jeszcze sporo do życzenia. Brak jest natomiast zupełnie programów dokonujących automatycznych przekładów pomiędzy językiem polskim a innymi dużymi i ważnymi językami używanymi w Europie, takimi jak choćby język francuski, hiszpański, włoski, niderlandzki, portugalski, szwedzki itp.

Z drugiej strony należy oczekiwać, że zapotrzebowanie na przekłady i tłumaczy będzie stale rosło zarówno w Polsce, jak i w Europie Zachodniej. Wystarczy sobie tylko uświadomić fakt, jak niezwykle skomplikowaną mozaiką etniczną (i w związku z tym również językową) jest Europa, w porównaniu np. z Ameryką Północną gdzie na większej części kontynentu w powszechnym użyciu jest język angielski (należy oczywiście uwzględnić w tym miejscu poprawkę na stale wzrastającą liczbę osób w Stanach Zjednoczonych używających języka hiszpańskiego). Przystąpienie Polski i innych krajów Europy Środkowej do Unii Europejskiej spowoduje, że organizacja ta stanie się prawdziwą Wieżą Babel, a zapotrzebowanie na tłumaczenie na wszystkie języki oficjalne państw członkowskich olbrzymiej ilości dokumentów i aktów prawnych może stać się naprawdę poważnym problemem. O tym, że bariery językowe stanowić mogą skuteczną zaporę przeciwstawiająca się integracji narodów Europy, świadczyć może wypadek jaki zdarzył się na początku 2001 roku w Belgii. Warto w tym miejscu wspomnieć, że Królestwo Belgii powstało w 1830 roku, jako państwo obejmujące terytorium zamieszkałe przez dwie narodowości: Wallonów (mówiących po francusku) i Flamandów (mówiących językiem niderlandzkim), przy czym Konstytucja Belgii

w sposób bardzo dokładny opisuje przebieg granicy językowej między dwiema wymienionymi narodowościami (stolica Bruksela jest dwujęzyczna). Właśnie bariera językowa, jaka istnieje pomiędzy dwoma narodami Belgii stała się przyczyną wspomnianego wypadku. Mianowicie, maszynista pociągu zmierzającego z obszaru francuskojęzycznego Belgii do jej obszaru niderlandzkojęzycznego wjechał na niewłaściwy tor. Fakt ten został zauważony przez dyżurnego stacji, który natychmiast zatelefonował do swojego kolegi z niderlandzkojęzycznej części Belgii i powiedział mu w swoim ojczystym języku (czyli po francusku), aby tamten nie wypuszczał ze swojej stacji pociągu jadącego w przeciwną stronę. Niestety w odpowiedzi usłyszał tylko: Ik begrijp het niet! (co w języku niderlandzkim znaczy: Nie rozumiem!), a następnie trzask odłożonej pośpiesznie słuchawki. W kilka minut później doszło do bardzo poważnej katastrofy kolejowej, w której było wiele ofiar śmiertelnych.

Mając na uwadze powyższą dygresję, można dobrze uświadomić sobie skalę rozważanego problemu i zobaczyć zagadnienia translacji maszynowej we właściwym świetle, bowiem translacja maszynowa dawno przestała być zagadnieniem czysto akademickim, a stała się palącą potrzebą chwili.

2. RÓŻNORODNOŚĆ JĘZYKÓW ŚWIATA

Okazuje się, że na pytanie ile jest na świecie języków, wcale nie jest łatwo odpowiedzieć. Autorowi książki [18] udało się zgromadzić 6821 lingwonimów, czyli nazw języków lub nazw tzw. etnolektów. Etnolekt jest bardzo wygodnym pojęciem stosowanym w lingwistyce i oznacza środek komunikacji językowej używany przez daną grupę etniczną. Etnolektem może być zatem równie dobrze język, jak i coś co uznawane jest jedynie za dialekt, narzecze lub gwarę. Zatem mówiąc o etnolekcie pomija się trudną często do rozstrzygnięcia kwestię, czy dany środek komunikacji językowej jest już odrębnym językiem, czy tylko dialektem jakiegoś innego języka.

O tym jak trudny jest to do rozstrzygnięcia problem, najlepiej świadczą chyba ciągnące się w nieskończoność spory na temat tego czy kaszubski jest odrębnym językiem (bardzo zresztą bliskim językowi polskiemu), czy też jest to jedynie dialekt języka polskiego [33]. Obecnie w językoznawstwie panuje raczej powszechny pogląd, w myśl którego kaszubskiemu przyznaje się status odrębnego języka, ale nie ma co do tego wśród badaczy jednomyślnej zgody. Podobne spory dotyczą również języków holenderskiego i flamandzkiego, które albo uznaje się za dwa odrębne języki, albo też określa się je wspólnym mianem języka niderlandzkiego, co jest obecnie w językoznawstwie dominująca tendencją [34]. Nietrudno wszakże wskazać kontrprzykład takiej unifikacji, ponieważ na przykład jeszcze kilka lat temu mówiono wyłącznie o języku serbsko-chorwackim, obecnie mówi się prawie wyłącznie o serbskim i chorwackim jako o odrębnych językach.

Zamieszczone powyżej przykłady pokazują, jak nieostrą i trudną do uchwycenia jest granica oddzielająca dwa blisko ze sobą spokrewnione języki. Zatem uznanie różnych etnolektów za jeden język bądź kilka odrębnych języków jest często sprawą arbitralną,

zależną często od argumentów pozalingwistycznych – głownie od sytuacji politycznej panującej w danym regionie. Na przykład język perski (Iran), język dari (Afganistan) oraz język tadżycki (Tadżykistan) są całkowicie zrozumiałe dla każdego z użytkowników dowolnego z tych języków, a mimo to oficjalnie są uznawane za odrębne języki. Podobna sytuacja występuje, miedzy innymi, w przypadku języków bułgarskiego (Bułgaria) i macedońskiego (Macedonia) oraz w przypadku języków tajskiego (Tajlandia) i laotańskiego (Laos). Z drugiej jednak strony istnieją dość duże opory przed przyznaniem odrębnych statusów wariantom języka niemieckiego używanym w Luksemburgu, Bawarii i Szwajcarii, w sytuacji gdy osoby mówiące po bawarsku, szwajcarsku bądź luksembursku mają duże problemy z porozumieniem się z osobami posługującymi się literackim językiem niemieckim. W tym kontekście dość śmieszne muszą wydawać się próby przyznania, swego czasu, odrębnego statusu niemczyźnie NRD (planowano podobno nawet reformę ortografii) [18]. Również język riukiuański (Wyspy Riukiu u wybrzeży Japonii) w Japonii uważany jest, głównie z powodów politycznych (na wyspach Riukiu istnieją tendencje separatystyczne), za dialekt języka Japońskiego, podczas gdy jest to język dla Japończyków całkowicie niezrozumiały. Co więcej, genetyczne pokrewieństwo języków japońskiego i riukiuańskiego nie zostało jeszcze ostatecznie przez badaczy potwierdzone. Z kolei mało znanym w Polsce faktem jest to, iż kilkadziesiąt procent ludności Włoch nie zna (bądź zna jedynie biernie) literackiego języka włoskiego (określanego jako język toskański), ponieważ na co dzień używa jednego z języków takich jak: lombardzki, liguryjski, piemoncki, kalabryjski, sycylijski, sardyński (choć właściwie należałoby mówić o czterech różnych językach sardyńskich), wenecki i emiliański. Języki te są w dużej mierze wzajemnie niezrozumiałe, natomiast poza Włochami uważa się, że Włosi – mówią po prostu po włosku [35]. Analogiczna sytuacja występuje również w Hiszpanii w regionach takich, jak Katalonia, Asturia i Galisja (oczywiście również w Kraju Basków, ale baskijski nie jest językiem romańskim).

Podobnie często mówi się o języku arabskim, jak gdyby był tylko jeden taki język, wspólny dla wszystkich mieszkańców państw arabskich, gdy tym czasem oprócz klasycznego języka arabskiego – będącego językiem Koranu, w powszechnym użyciu jest cała masa mówionych języków arabskich, takich jak: egipski, syryjski, jordański, libański, iracki, omański, saudyjski, libijski, tunezyjski, algierski, marokański, mauretański i maltański, które są często wzajemnie niezrozumiałe [36]. Podobnie wygląda sytuacja z językiem chińskim, który właściwie stanowi niekończącą się, pogmatwaną sieć różnorodnych dialektów o niezwykle skomplikowanych wzajemnych powiązaniach.

Na podstawie zamieszczonych powyżej przykładów widać, że na pytanie ile jest języków na świecie, praktycznie nie można udzielić dokładnej odpowiedzi. Amerykański Center for Applied Linguistics z siedzibą w Waszyngtonie zgromadził około 18000 nazw języków. Jest jednak sprawą oczywistą, że podana liczba nie odpowiada liczbie języków używanych współcześnie na świecie, ponieważ wiele z wymienionych języków uważa się już za wymarłe (np. sanskryt, sumeryjski, akadyjski, mozarabicki, koptyjski itp.), wiele z języków może występować pod kilkoma różnymi nazwami oraz pewne nazwy mogły trafić na rozważaną listę po prostu przez pomyłkę bądź zwykłą niekom-

petencję, która w konfrontacji z tak olbrzymim i różnorodnym materiałem jest najzwyczajniej nieunikniona.

Języki świata są przez badaczy klasyfikowane, przy czym najbardziej popularna jest ich klasyfikacja genetyczna, uwzględniająca wzajemne podobieństwa języków ze względu na ich pochodzenie od wspólnego praprzodka – prajęzyka. Wśród badaczy nie ma powszechnie przyjętej – jedynie słusznej – klasyfikacji języków świata. Zamieszczona poniżej klasyfikacja, która pod wieloma względami wydaje się być atrakcyjną, została podana za książką [18]. Według tej klasyfikacji języki świata dzieli się na 27 rodzin językowych, do których można zaliczyć:

- Języki indoeuropejskie
- Języki semito-chamickie
- Języki uralskie
- Języki ałtajskie
- Języki koreański, japoński i riukiuański
- Języki kaukaskie
- Języki drawidyjskie
- Języki paleoazjatyckie
- Języki eskimosko-aleuckie
- Języki austroazjatyckie
- Języki chińsko-tybetańskie
- Języki austronezyjskie
- Języki papuaskie
- Języki australijskie
- Języki afrykańskie zindż
- Języki khoisan
- Języki algonkińskie, saliskie, chimakańskie, wakaszańskie
- Języki na-dene
- Języki siouańskie
- Języki hokańskie
- Języki penutiańskie
- Języki jukijskie
- Języki oto-mangueskie
- Języki ge-pano-karaibskie
- Języki czibczańskie
- Języki aztecko-tanoańskie
- Języki andyjsko-równikowe

Oprócz wymienionych powyżej rodzin językowych występują również tzw. języki izolowane, których nie można zaliczyć do żadnej rodziny językowej, czasami dlatego, że inni reprezentanci takiej rodziny należą już do języków martwych, które już dawno temu odeszły niezauważone do historii, albo też obecny stan badań lingwistycznych nie pozwala na wysnucie przekonujących argumentów za przynależnością danego języka do takiej czy innej rodziny językowej. Z bardziej znanych języków uznawanych obecnie za izolowane można wymienić język ajnuski (był w użyciu jeszcze do początków

XX wieku na Wyspach Kurylskich, na Kamczatce i na Hokaido – w jego zbadanie znaczący duży wkład wniósł zesłany na Daleki Wschód Bronisław Piłsudski, brat marszałka Józefa Piłsudskiego), język baskijski oraz starożytne języki: etruski, hurycki (Mezopotamia II tysiąclecie p.n.e.) i sumeryjski.

Z kolei indoeuropejska rodzina językowa, do której należy również język polski, dzieli się na następujące podrodziny [18]: indo-irańska, anatolijska, tocharska, armeńska, albańska, grecka, germańska, słowiańska, bałtycka, celtycka i italska.

Do podrodziny słowiańskiej należą następujące języki [18]: polski, kaszubski, czeski, słowacki, dolnołużycki, górnołużycki, połabski (wymarły), dialekty Meklemburgii (wymarłe), strocerkiewnosłowiański (wymarły), bułgarski, banacki (używany na terytorium Rumunii i uważany czasami za dialekt języka bułgarskiego), macedoński, serbochorwacki (obecnie mówi się raczej o odrębnych językach serbskim i chorwackim), słoweński, staroruski (wymarły), rosyjski, ukraiński, rusiński (bliski ukraińskiemu i używany w Wojewodinie i Chorwacji) i białoruski.

Oprócz klasyfikacji genetycznej, grupującej języki ze względu na ich pochodzenie wyróżnia się również inne klasyfikacje porządkujące języki świata ze względu na cechy ich budowy. Do najczęściej spotykanych typów klasyfikacji języków świata można zaliczyć klasyfikację fonologiczną, morfologiczną, syntaktyczną, semantyczną, a także klasyfikację systemów graficznych języków [18]. Niestety nie ma tutaj miejsca na dokładniejsze rozważenie poruszonych w tym punkcie zagadnień, można jedynie powiedzieć, że w świetle wymienionych powyżej klasyfikacji świat języków jawi się jako niezwykle liczny, bogaty i zdumiewająco różnorodny. W związku z tym w dającej się przewidzieć przyszłości bezrobocie tłumaczom i lingwistom raczej nie grozi (zwłaszcza, że znakomita większość języków używanych obecnie na świecie – ponad 80%, pozostaje jeszcze nie zbadana). Wręcz przeciwnie, tłumacze już teraz są zasypywani nadmiarem pracy, a tłumaczenie zwłaszcza z nietypowych języków (jak np. język japoński) jest bardzo kosztowne. Widać zatem, że prowadzenie prac nad translacją maszynową ma sens, ponieważ może przyczynić się do znacznej redukcji kosztów i czasu potrzebnego na dokonanie przekładu. Czy jednakże kontynuowanie prac nad translacja maszynową nie doprowadzi kiedyś do powszechnego bezrobocia w środowisku tłumaczy? Jest to dylemat analogiczny do sytuacji panującej w dziewiętnastym wieku, kiedy to obawiano się, że powszechne wprowadzenie do produkcji maszyn spowoduje masowe zwolnienia robotników z fabryk i wywoła bezrobocie na niespotykaną skalę. Na ile obawy związane z translacją maszynową są uzasadnione powinno wyjaśnić się już w następnym punkcie, w którym rozważony zostanie dokładnie proces translacji (nie tylko maszynowej) pomiędzy dwoma językami naturalnymi.

3. JAK WIELE MOŻNA PRZETŁUMACZYĆ?

W niniejszym punkcie naświetlone zostaną podstawowe trudności, problemy i ograniczenia, na jakie nieuchronnie natrafiają twórcy systemów automatycznej translacji. Dyskusję na ten temat należy zacząć w ogólności od stwierdzenia, że dokonywanie

przekładów z języka obcego (nawet przez człowieka znającego biegle oba języki) nie jest sprawą bynajmniej łatwą. Czytając dowolną książkę przełożoną z języka obcego zdecydowana większość czytelników nie zdaje sobie nawet sprawy z faktu, jak wiele pracy i trudu musiał włożyć tłumacz, aby uzyskać obraz świata takim, jakim opisał go autor [16]. Okazuje się bowiem, że użytkownicy różnych języków często w zupełnie odmienny sposób opisują otaczający ich świat oraz inne elementy tego świata uznają za istotne i zasługujące na zawarcie informacji o nich w formułowanych przez siebie wypowiedziach [17]. Działalność językowa człowieka polega bowiem na charakteryzowaniu sytuacji (rzeczywistych bądź hipotetycznych), ich opisywaniu lub formułowaniu sądów na ich temat. Każda taka sytuacja jest niezwykle skomplikowana i składa się na nią praktycznie nieskończona ilość elementów składowych – zatem zawarcie w wypowiedzi pełnej informacji o charakteryzowanej sytuacji nie wydaje się być możliwe. Człowiek w swej działalności językowej skazany jest z konieczności na wybór tylko nielicznych elementów opisujących sytuację, w której przyszło mu w jakimś stopniu uczestniczyć. Elementy takie powinny być najistotniejsze i w możliwie najlepszy sposób charakteryzować opisywaną sytuację. Jednakże badania prowadzone w dziedzinie lingwistyki porównawczej dowodzą niezbicie, że bardzo często użytkownicy różnych języków, postawieni w obliczu dokładnie takiej samej sytuacji, uznają za istotne niekoniecznie te same elementy [18]. Zagadnienie to zostanie najlepiej zilustrowane na kilku następujących przykładach.

Na przykład chcąc przetłumaczyć na język polski wyrwane z kontekstu węgierskie zdanie

Ő tanár.

tłumacz napotyka dość sporą trudność, ponieważ ma do wyboru dwie możliwości

Jest nauczycielem.

lub

Jest nauczycielką.

Język węgierski bowiem nie zna pojęcia rodzaju gramatycznego i nie wiedząc o kim jest mowa, nie można takiego zdania poprawnie na język polski przetłumaczyć. W tym przypadku wypowiedź w języku polskim zawiera więcej informacji – widocznie z jakiś powodów dla użytkowników języka polskiego w przypadku opisu rozważanej sytuacji ważna jest informacja o płci osoby, której wypowiedź dotyczy.

Podobna trudność występuje podczas próby przetłumaczenia szwedzkiego zdania

Jag har tvättat bilen.

które również może zostać przełożone na dwa sposoby w zależności od płci osoby mówiącej:

Umyłem samochód.

bądź

Umyłam samochód.

W tym przypadku wypowiedź w języku polskim również okazuje się być bogatsza w informacje.
Z kolei podając polski ekwiwalent angielskiego pytania (skierowanego do jednej osoby):

Are you happy?

tłumacz posiada do wyboru aż cztery możliwości, mianowicie:

Czy jesteś szczęśliwy?

Czy jesteś szczęśliwa?

Czy pan jest szczęśliwy?

Czy pani jest szczęśliwa?

W tym przypadku oprócz informacji o płci osoby konieczna jest również informacja o wzajemnych relacjach społecznych pomiędzy osobami uczestniczącymi w rozmowie. Niestety informacje takie, jak widać istotne dla użytkownika języka polskiego, nie są zawarte w wypowiedzi osoby posługującej się językiem angielskim. Dla kontrastu, problem taki nie występuje w języku francuskim, gdzie każde z powyższych zdań posiada swój odrębny odpowiednik:

Est-ce que tu és hereuer?

Est-ce que tu és hereuse?

Est-ce que vous êtes hereuer?

Est-ce que vous êtes hereuse?

W tym wypadku można powiedzieć, że język francuski i polski w opisie tej samej sytuacji zawierają jednakową ilość informacji.

Na podstawie zamieszczonych powyżej przykładów mogłoby się wydawać, że język polski jest jakimś specyficznym językiem, którego wypowiedzi są bardziej bogate niż wypowiedzi formułowane w innych językach. O tym, że jednak nie zawsze jest to prawdą można przekonać się podczas próby przekładu na język hiszpański polskiego zdania:

My mamy szczęście.

które może zostać przetłumaczone jako:

Nosotros tenemos suerte.

albo

Nosotras tenemos suerte.

W zależności od tego czy zdanie to jest wypowiadane przez mężczyzn (w przypadku pierwszym), czy przez kobiety (w przypadku drugim). W tej sytuacji to właśnie język hiszpański, a nie język polski przekazuje informację o płci mówiących osób. Posiadanie odmiennych form dla zaimków osobowych „my" (nosotros i nosotras) oraz „wy" (vosotros i vosotras) jest interesującą osobliwością języka hiszpańskiego, niespotykaną w innych językach romańskich, takich jak np. portugalski, francuski i włoski.

Jednakże na jeszcze większe osobliwości można natrafić analizując paralelne wypowiedzi zapisane w językach bardzo odległych od siebie ze względu na ich klasyfikację genetyczną, geograficzną i typologiczną [18].

Na przykład chcąc przetłumaczyć na język japoński zdanie:

Brat pani Huang jest też nauczycielem.

trzeba wiedzieć czy ów brat jest od pani Huang młodszy, czy też straszy. Bowiem w pierwszym przypadku przekład będzie miał postać:

Fuangu-san-no otootosan-mo sensei-desu.

natomiast w drugim:

Fuangu-san-no oniisan-mo sensei-desu.

Podobnie tłumacząc polskie słowo „ubierać" na język japoński tłumacz ma do wyboru całą gamę ekwiwalentów, w zależności od tego, czy ubierane mają być: buty, okulary, kapelusz, rękawiczki, płaszcz, czy też szalik. Bowiem w zależności od rodzaju przedmiotu, który ma być ubierany japońskie tłumaczenie czasownika „ubierać" będzie następujące:

haku (ubierać buty)

kakeru (ubierać okulary)

kaburu (ubierać kapelusz)

hameru (ubierać rękawiczki)

haoru (ubierać płaszcz)

shimeru (ubierać szalik)

Z kolei w celtyckim języku walijskim wyraz „glas" można przetłumaczyć na polski jako zielony, niebieski bądź szary, ponieważ język walijski nie posiada odrębnych słów na odróżnianie tych kolorów. Posiada natomiast, co może być zaskakujące, odrębne wyrazy pozwalające na odróżnienie różnych odcieni koloru zielonego, który w zależności od intensywności barwy określany jest słowem „glas" lub „gwyrdd".

Podobnie tłumacząc polskie słowo „iść" na język Indian Navaho, trzeba widzieć, czy porusza się jedna osoba, dwie osoby, kilka osób, czy też duża grupa ludzi, ponieważ w każdym z tych przypadków w języku navaho należy użyć innych zwrotów językowych. W języku navaho nie ma bowiem czasownika, któryby wszystkie wymienione powyżej sytuacje uogólniał i oznaczał czynność chodzenia bez uwzględniania liczby zaangażowanych w to osób. Analogiczna sytuacja występuje przy próbie tłumaczenia z języka angielskiego na polski angielskiego czasownika „to go", który w zależności od kontekstu może oznaczać iść, jechać, płynąć, przemieszczać się itp.

Zwracając się w języku angielskim do osoby o imieniu *John* można użyć, w zależności od stopnia poufałości z tą osobą zwrotu John! lub Johnny! Natomiast język polski jest pod tym względem znacznie bogatszy i tłumacz musi zdecydować się na którąś z następujących form: Janie!, Janku!, Janeczku!, Jasiu!, Jasieńku! itp. Podobnie sytuacja wygląda w języku czeskim, gdzie tłumacz ma do wyboru formy: Jane!, Jene!, Honzo!, Honziku! lub Honzičku! O tym jednak, którą z tych form wybrać musi zadecydować sam tłumacz, ponieważ oryginalny tekst w języku angielskim nic na ten temat nie mówi.

W języku koreańskim stopień poufałości pomiędzy osobami uczestniczącymi w rozmowie określany jest za pomocą specjalnych przyrostków dołączanych do czasowników, przy czym istnieje aż sześć typów takich przedrostków (sześć różnych stopni grzeczności). Należy zwrócić uwagę, że tłumacząc np. z języka angielskiego, w którym do dowolnej osoby zwracamy się za pomocą słowa „you", powstaje dość znaczny problem z wyborem odpowiedniej formy grzecznościowej, przy czym problemu tego nie da się w żaden sposób uniknąć, ponieważ na którąś z tych form (dobór odpowiednich przyrostków) trzeba się przecież zdecydować.

W Norwegii uczniowie zwracają się do swoich nauczycieli po imieniu. Zatem tłumacząc dosłownie z języka norweskiego na polski można na przykład otrzymać następujące zdanie:

Andrzeju, co będzie dzisiaj na zadanie domowe?

Jednakże czytelnik polski napotykając takie zdanie może potraktować je jako zbyt bardzo poufałe i wyciągnąć na tej podstawie fałszywy wniosek o braku kultury osobistej wśród norweskich uczniów. Z kolei tłumacząc takie zdanie jako

Panie profesorze, co będzie dzisiaj na zadanie domowe?

nie uwzględnia się pewnego stopnia wzajemnej bliskości i zażyłości, jaki panuje w norweskiej szkole. Zatem w pewnym sensie rozważane zdanie jest nieprzekładalne, ponieważ poszczególne społeczności językowe zamieszkują jak gdyby dwa odrębne światy. Na podstawie podobnych przykładów amerykański teoretyk przekładu Lawrence Venuti stworzył dwa terminy, dość trafnie opisujące postępowania tłumacza wobec sytuacji językowych, które z wielkim trudem poddają się przekładowi, są to tzw. uzagranicznienie (ang. *foreignization*) i udomowienie (ang. *domestication*). Pierwsza możliwość polega jak gdyby na wysłaniu czytelnika za granicą i pokazaniu mu innej kultury, druga zaś polega na zredukowaniu obcego tekstu do ram języka przekładu, czyli na sprowadzeniu autora do kraju czytelnika [16].

Chcąc przetłumaczyć angielskie zdanie:

I drop it.

na język polski, tłumacz nie ma większych kłopotów – po prostu powie

Upuszczam to.

Natomiast chcąc przetłumaczyć to samo zdanie na język Indian Navaho trzeba jeszcze znać stopień zaangażowania osoby mówiącej w wykonywaną czynność oraz wiedzieć co będzie upuszczane, bowiem dobór użytego czasownika zależy od rodzaju upuszczanego przedmiotu, np. od tego czy jest on okrągły, czy płaski, czy też podłużny, miękki lub twardy itp.

Analogiczna sytuacja występuje w języku japońskim [26]. Na przykład chcąc przetłumaczyć na język japoński liczebnik *dwa* tłumacz ma do wyboru wiele możliwości. Jeżeli liczebnik „dwa" dotyczy dwóch długich obiektów, wówczas należy użyć japońskiego słowa „ni-hon", jeżeli są to obiekty płaskie wówczas należy użyć słowa „ni-mai". Natomiast dwoje ludzi w języku japońskim brzmi „futari".

Pewną osobliwością niektórych języków jest występowanie oprócz form rzeczowników, właściwych dla liczby pojedynczej i mnogiej, również tzw. liczby podwójnej, odnoszącej się do dwóch egzemplarzy danego obiektu. Jako ciekawostkę można podać, że liczba podwójna występowała również w języku polskim – jeszcze do około XVI wieku (dla przykładu można wymienić formy liczby pojedynczej, podwójnej i mnogiej następujących rzeczowników: oko, oka, oczy lub ucho, ucha, uszy) [33]. Obecnie taką osobliwość językową można zaobserwować jeszcze w języku słoweńskim, gdzie np.:

- lipa – to „lipa" (l. pojedyncza), „lipi" (l. podwójna) i „lipe" (l. mnoga)
- brzytwa – to „britev" (l. pojedyncza), „britvi" (l. podwójna) i „britve" (l. mnoga)
- brat – to „brat" (l. pojedyncza), „brata" (l.podwójna) i „brati" (l. mnoga)
- miasto – to „mesto" (l. pojedyncza), „mesti" (l. podwójna) i „mesta" (l.mnoga)

Zatem tłumacz, chcąc przetłumaczyć na język słoweński polski rzeczownik „miasta" staje przed niełatwym wyborem, ponieważ musi wiedzieć ile tych miast jest. Bowiem, jeśli były tylko dwa miasta, należy użyć słoweńskiego rzeczownika „mesti", jeśli natomiast było ich więcej należy użyć wyrazu „mesta".

Podobne przykłady można mnożyć chyba w nieskończoność. Pokazują one jak niezwykle trudną jest praca tłumacza ze względu na ograniczenia systemowe, które tkwią w każdym z języków naturalnych. Tłumacze często powtarzają, że nie można nauczyć się języka obcego na tyle dobrze, żeby tłumaczyć literaturę piękną – i jest w tym stwierdzeniu zapewne sporo prawdy [16]. Przytoczone powyżej przykłady rozwiewają pewien mit o łatwości tłumaczenia, które w obiegowej opinii polega jedynie na odnajdywaniu w słowniku ekwiwalentów wyrazów, ich podstawianiu w miejsca wyrazów występujących w tekście oryginalnym i następnie uzgadnianiu ich formy zgodnie z regułami języka, na który się dany teks tłumaczy. Gdyby tak było naprawdę, wówczas translacja maszynowa byłaby dziecinnie prosta i nie byłby to temat wart badań naukowych. Niestety, ze względu na różnice w funkcjonowaniu poszczególnych języków, z których każdy jest systemem o odmiennej budowie i sposobie działania, translacja maszynowa okazuje się być dziedziną niezwykle trudną, wymagającą prowadzenia poważnych badań interdyscyplinarnych, łączących informatykę z lingwistyką i filozofią języka. Również zawodowy tłumacz musi bardzo często wykazać się pomysłowością, talentem i niemałym polotem, czyli tymi cechami ludzkiego umysłu, których nie da się żadnym sposobem sprowadzić do realizacji algorytmu przez maszynę cyfrową. Aby tłumaczyć literaturą piękną potrzebna jest po prostu inteligencja prawdziwa – ludzka, a nie sztuczna – komputerowa. Można zatem postawić pytanie czy istnieje w ogóle szansa na całkowite zastąpienie człowieka tłumacza przez komputer [19]? Wydaje się, że przy obecnym stanie techniki komputerowej nie. Zatem prawdopodobnie nigdy nie będziemy mieli systemu komputerowego potrafiącego tłumaczyć automatycznie na język polski dzieła Szekspira, Dantego lub Homera. Nie powinno to jednak nikogo zbytnio martwić, ponieważ brak takiej możliwości wcale jeszcze nie przesądza o tym, że translacja maszynowa jest bezużyteczna. Zresztą zarzut taki można trafnie porównać do zarzutu, jaki ktoś mógłby postawić wobec robotów przemysłowych, że nie potrafią np. zatańczyć baletu Jezioro Łabędzie Piotra Czajkowskiego. Owszem nie mogą – i prawdopodobnie sztuka konstruowania robotów nigdy nie osiągnie takiej precyzji, aby roboty przemysłowe mogły uzyskać wymaganą do tego gracje, zwinność, sprawność i płynność ruchów – ale przecież nie w tym celu konstruowane są roboty przemysłowe. Brak tak precyzyjnych możliwości ruchowych wcale nie oznacza braku przydatności robotów do realizacji wielu zadań spotykanych w praktyce przemysłowej, jak np. montaż elementów, spawanie, zgrzewanie elementów konstrukcji, szlifowanie powierzchni itp. Podobnie zarzut, że systemy translacji automatycznej nigdy nie będą w stanie tłumaczyć literatury pięknej jest chybiony i całkowicie nie na miejscu, bo nie w tym celu systemy te są opracowywane [20]. Tłumaczenie maszynowe może okazać się natomiast bardzo przydatne i częstokroć wręcz nieodzowne w wielu międzynarodowych korporacjach, które posiadają swoje oddziały rozrzucone po całym świecie i muszą w związku z tym tłumaczyć olbrzymie ilości dokumentów prawnych, handlowych i technicznych na wiele języków – co w warunkach powszechnej globalizacji gospodarki staje się codziennością. Również przedsiębiorstwa, które eksportują swoje produkty za granice do wielu różnych krajów i obszarów językowych stają często przed koniecznością tłumaczenia na mnóstwo języków dołączanych do tych produktów ulotek

informacyjnych, instrukcji obsługi, podręczników użytkowania, katalogów, dokumentacji technicznych itp. Właśnie w takich sytuacjach ograniczenia tłumaczonego materiału leksykalnego do pewnej wąskiej dziedziny przedmiotowej (np. ulotki na chemicznych środkach czystości, instrukcje obsługi sprzętu RTV, podręczniki użytkowania samochodów itp.) translacja maszynowa pozwala na odniesienie dużego sukcesu, w postaci uzyskania przekładu o prawie stuprocentowej dokładności, wymagającego później jedynie znikomej, wręcz kosmetycznej interwencji człowieka tłumacza, celem dokonania potrzebnych zmian i poprawek.

Niestety stworzenie perfekcyjnie działającego (nawet w ograniczeniu do wybranej dziedziny) systemu automatycznej translacji nie jest sprawą łatwą, a do rozwiązania pozostaje wciąż jeszcze wiele problemów. Co zatem sprawia największą trudność systemom translacji maszynowej? Odpowiedź na to pytanie zostanie udzielona w kolejnym punkcie na podstawie analizy licznych przykładów.

3. CO SPRAWIA NAJWIĘCEJ TRUDNOŚCI W TRANSLACJI AUTOMATYCZNEJ?

Zamieszczone w poprzednim punkcie przykłady wykazały dobitnie, że zadanie translacji nie jest łatwe nawet dla zawodowego tłumacza, który często w celu dokonania przekładu musi wykazać się niemałą pomysłowością i inwencja twórczą oraz często musi korzystać również z pewnych informacji pozalingwistycznych dotyczących realiów sytuacji, której dotyczy rozmowa. W przypadku translacji maszynowej występuje jeszcze inna dodatkowa trudność, którą jest wieloznaczność charakterystyczna dla wypowiedzi formułowanych w każdym (bez wyjątku) z języków naturalnych. Często człowiek tłumacz nie ma większych problemów z taką wieloznacznością, ponieważ wykorzystując swoją wiedzę o otaczającej go rzeczywistości oraz zdrowy rozsadek potrafi, w naturalny sposób, taką wieloznaczność rozwikłać, wybierając właściwe znaczenie usłyszanej wypowiedzi. Okazuje się jednak, że w przypadku zastąpienia człowieka tłumacza komputerem, sprawa nie jest już taka prosta, gdyż to co pozornie dla człowieka jest oczywiste może okazać się bardzo trudne lub wręcz niemożliwe do ujęcia w ramy wykonywanego przez komputer algorytmu. Bo niby jak można stworzyć algorytm symulujący zdrowy rozsadek człowieka?

Wieloznaczność wypowiedzi jest immanentną cechą każdego języka naturalnego. W dowolnym języku naturalnym zwykle trudno jest wskazać słowo, które nie miałoby co najmniej dwóch różnych znaczeń. W takim przypadku mówi się o wieloznaczności leksykalnej (ang. *lexical ambiguity*). Często okazuje się również, że całe wypowiedzi lub zdania mogą być interpretowane na wiele różnych sposobów. Mówi się wówczas o wieloznaczności strukturalnej (ang. *structual ambiguity*). O tym, że wieloznaczność występująca powszechnie w każdym języku naturalnym i w związku z tym może okazać się zmorą translacji maszynowej, przekonać się można na kilku następujących przykładach.

Mianowicie, angielskie zdanie:

Put the paper in the printer. Then switch it on.

program automatycznej translacji może przetłumaczyć na język polski na dwa alternatywne sposoby:

Wsadź papier do drukarki. A następnie włącz go.

albo też:

Wsadź papier do drukarki. A następnie włącz ją.

w zależności od tego, czy angielski zaimek „it" dotyczy słowa „paper" (papier), czy też słowa „printer" (drukarka). Oczywiście człowiek tłumacz nie będzie miał w tym wypadku najmniejszych problemów, ponieważ wykorzystując swoją wiedzę o świecie oraz posługując się zdrowym rozsądkiem, będzie wiedział, że włącza się drukarkę, a nie papier, ale skąd o tym może wiedzieć komputer? Oczywiście można próbować (i tak się też czyni) wyposażyć komputer w odpowiednią bazę danych, w której zawarte by były tego typu informacje, powstaje jednak pytanie, czy w takiej bazie danych można ująć i przewidzieć wszystkie potencjalne sytuacje, jakie mogą zaistnieć w świecie rzeczywistym?

Podobny problem występuje podczas próby automatycznego przetłumaczenia zdania:

Cleaning fluids can be dangerous.

W zdaniu tym angielskie słowo „cleaning" może zostać potraktowane zarówno jako rzeczownik odczasownikowy lub jako przymiotnik. Jeżeli zostanie potraktowane jako rzeczownik wówczas translacja rozważanego zdania będzie miała postać następującą:

Czyszczenie płynów może być niebezpieczne.

Natomiast w przypadku potraktowania tego słowa jako przymiotnik otrzymamy:

Czyszczące płyny mogą być niebezpieczne.

Aby wybrać właściwą z tych dwóch możliwości potrzebna jest wiedza na temat dziedziny, której dana wypowiedź dotyczy. Oczywiście wiedzę taką może posiadać człowiek tłumacz, natomiast wyposażenie w nią komputera może okazać się bardzo trudne.

Często o wyborze danej możliwości decyduje kontekst wypowiedzi oraz do jej właściwego zrozumienia potrzebne są także pewne informacje pozalingwistyczne, np. konieczne jest aby osoby uczestniczące w rozmowie widziały opisywaną sytuację. Na przykład bez takich dodatkowych informacji przetłumaczenie zdania:

I see a man with a telescope.

jest niemożliwe nawet dla człowieka tłumacza, ponieważ do wyboru są dwie równoprawne możliwości:

Widzę człowieka z teleskopem.

albo

Widzę człowieka za pomocą teleskopu.

W takiej sytuacji dopiero zobaczenie tego człowieka (a właściwie stwierdzenie faktu czy posiada on teleskop, czy też nie) może ostatecznie rozwiać wątpliwości.
Podobne trudności sprawia zdanie:

Attach the printer to the PC with a parallel interface cable

które może zostać przetłumaczone jako:
Dołącz drukarkę do komputera PC za pomocą kabla interfejsu równoległego.
lub

Dołącz drukarkę do komputera PC z kablem interfejsu równoległego.

Człowiek tłumacz, posiadający dostateczną wiedzę na temat komputerów nie będzie miał większego problemu z opowiedzeniem się za pierwszą z tych możliwości. Wiedział będzie bowiem, że komputer łączy się z drukarką za pomocą specjalnego kable interfejsu równoległego, natomiast komputerów PC, które byłyby jakoś zintegrowane z takim kablem po prostu nie ma i w związku z tym druga możliwość pozbawiona jest sensu. Jakim sposobem jednak można wyposażyć w tego typu wiedzę program translacji automatycznej?

Podobny problem powstaje przy próbie przetłumaczenia angielskiego wyrażenia:

a piece of glass and metal

które może zostać przełożone na dwa alternatywne sposoby, jako:

kawałek szkła i metal

lub

kawałek szkła i metalu

Aby zadecydować o wyborze którejś z tych możliwości osoba tłumacząca takie wyrażenie musi ten przedmiot po prostu widzieć albo też musi wiedzieć, o czym jest mowa. Jeżeli rozważany jest przedmiot, który stanowi zlepek szkła i metalu o niewiel-

kich rozmiarach, wówczas zostanie wybrana druga z podanych możliwości. Jeżeli natomiast mamy do czynienia z oddzielnymi obiektami, z których pierwszy jest ze szkła a drugi z metalu, wówczas należy wybrać wariant pierwszy. Jednak w języku angielskim obie te możliwości wyrażane są w identyczny sposób – natomiast w języku polskim już nie.

Po powyższych uwagach nie trzeba chyba wspominać, jak wiele problemów sprawia dla programów translacji automatycznej tłumaczenie idiomów, czyli wypowiedzi, których znaczenia nie można odczytać analizując bezpośrednio znaczenia wchodzących w ich skład słów. Na przykład w potocznym języku amerykańskim zdanie:

Sam kicked the bucket.

znaczy po prostu, że ów Sam umarł (wyrażając się niezbyt poprawnie po polsku można by powiedzieć przez analogię, że „Sam kopnął w kalendarz"). Z drugiej jednak strony zdanie takie można by równie dobrze przetłumaczyć jako:

Sam kopnął wiadro.

co też może być poprawnym tłumaczeniem, bo przecież Sam mógł rzeczywiście kopnąć stojące na jego drodze i zawadzające mu wiadro. O tym, która z możliwości jest poprawna decyduje zatem kontekst wypowiedzi i sytuacja, w której ta wypowiedź została wygłoszona – czyli pewne informacje pozajęzykowe.

W świetle powyższych rozważań wydaje się zatem, że do tego, aby można było dokonywać przekładu zdań dotyczących ogólnej dziedziny działalności ludzkiej potrzebna jest pewna doza inteligencji i zdrowego rozsądku. W ostatnim czasie bardzo wiele mówi się w informatyce o systemach sztucznej inteligencji. Powstaje zatem pytanie, czy systemy te są w stanie osiągnąć poziom inteligencji porównywalny z inteligencją człowieka? Czy też słowo inteligencja zostało dopisane do nazwy tych systemów niejako na wyrost? Powstaje zatem pytanie, czy prowadzenie badań nad translacje maszynową posiada w takiej sytuacji jeszcze jakikolwiek sens?

Uwzględniając obecne sukcesy w zastosowaniach systemów automatycznej translacji odpowiedź na tak postawione pytanie jest prosta. Badania nad translacja maszynowa są słuszne, potrzebne i prowadzone są we właściwym kierunku, o ile dotyczą języków kontrolowanych, które ograniczają się do pewnej wybranej dziedziny działalności językowej człowieka [40]. Jednakże prowadzenie badań nad w pełni automatyczną translacją maszynową o wysokiej jakości (ang. *fully-automatic high-quality machine translation*), działającą dodatkowo na nieograniczonych dziedzinowo tekstach (ang. *unrestricted text*) wydaje się za tym być mrzonką przy obecnym stanie technologii informatycznej [39], [43]. Wielu badaczy uważa, że aby mógł nastąpić tutaj jakiś przełom dokonać musi się wcześniej potężna rewolucja w fizyce, która doprowadzi do powstania zupełnie nowych metod przetwarzania informacji [27], [37]. Kiedy jednak taki przełom nastąpi i jak będzie wyglądał (może komputery kwantowe?) – nie sposób odpowiedzieć [28], [29]. Jego dokonanie się jest jednak konieczne, jeżeli w translacji

automatycznej ma wystąpić znaczący postęp, prowadzący do całkowitego zastąpienia człowieka tłumacza przez maszynę [38]. Bez jakiegoś przełomowego odkrycia w fizyce, przy użyciu komputerów można będzie tłumaczyć jedynie napisane w języku kontrolowanym, ograniczone do wybranej dziedziny teksty – wszak żarówkę elektryczną wynaleziono nie dlatego, że prowadzone były badania nad świeczką [19]!

Nie mniej jednak trzeba przyznać, że istniejące obecnie systemy translacji maszynowej okazują się bardzo przydatne, ponieważ ułatwiają pracę wielu osobom zatrudnionych w wielonarodowych korporacjach, a efekty ich działań dają się przełożyć na konkretne pieniądze [30]. Zatem badania nad istniejącymi systemami translacji automatycznej będą w przyszłości prowadzone, a ich doskonalenie i ich adaptacja do nowych dziedzin będą zapewne jeszcze długo kontynuowane [31]. Warto chyba w tym miejscu poświęcić chwilę czasu na zapoznanie się z historią systemów translacji automatycznej, ponieważ nie jest to dziedzina znowu tak młoda, jak mogło by się wydawać. Wręcz przeciwnie, translacja maszynowa posiada już dość długą i burzliwą – można nawet powiedzieć wręcz dramatyczną – historię, która zostanie naświetlona w kolejnym punkcie.

4. TROCHĘ HISTORII

Początki translacji maszynowej wywodzą się z technik kryptograficznych, które rozwinęły się bardzo mocno w latach Drugiej Wojny Światowej (chodziło oczywiście o szyfrowanie wiadomości i rozkazów wojskowych, aby były one niezrozumiałe w przypadku ich przechwycenia przez wroga). Na pomysł, że translacja może mieć coś wspólnego z kryptografią wpadli jako pierwsi dwaj amerykanie: Andrew Booth i Warren Weaver, którzy w 1949 roku skierowali w tej sprawie specjalne memorandum do Rockefeller Foundation, w którym postulowali podjęcie badań w tej dziedzinie [20]. Ich rozumowanie było następujące: „Załóżmy, że mamy przed sobą tekst napisany w języku rosyjskim, ale my uparcie będziemy twierdzić, że tak naprawdę tekst ten został napisany w języku angielskim, z tym, że następnie został zaszyfrowany przy użyciu pewnych dziwnych symboli. Zatem wszystko, co jest potrzebne do odczytania tego tekstu w języku angielskim to tylko znajomość szyfru – sposobu jego zakodowania."

Oczywiście później okazało się, że tego typu rozumowanie było zbyt wielkim uproszczeniem. Każde kodowanie (nawet najbardziej skomplikowane) polega na wzajemnie jednoznacznym podstawianiu elementów (ang. *one-for-one substitution process*), w związku z czym istnieje zawsze dokładnie jedno rozwiązanie, podczas gdy tłumaczenie jest procesem o wiele bardziej złożonym i wymagającym zaangażowania ludzkiej inteligencji. Z ograniczenia tego zdawali sobie sprawę również autorzy skierowanego do Rockefeller Foundation memorandum i w kilka lat później Warren Weaver skierował kolejne memorandum zawierające bardziej dojrzały naukowo pogląd na zagadnienie automatycznej translacji. Pojawienie się memorandum Weavera wywołało w świecie naukowym duże poruszenie i już w latach pięćdziesiątych istniało w Stanach Zjednoczonych i w Europie wiele ośrodków, w których podjęto badania w tym kierunku.

Jako ciekawostkę można podać, że rząd Wielkiej Brytanii wyasygnował w tym czasie na badania nad translacja automatyczną kwotę równą 20 milionów funtów. Po pierwszych sukcesach, jakie przyniosły podjęte badania przyszło jednak pewne rozczarowanie, ponieważ poniesione duże nakłady finansowe nie chciały się zwrócić w postaci konkretnych rynkowych produktów – spowodowane to było głównie ograniczeniami technologii komputerowej w tym czasie. Poza tym zaczęły pojawiać się pewne głosy teoretyków, którzy wątpili w ogóle w możliwość zautomatyzowania procesu translacji – przynajmniej przy stanie ówczesnej wiedzy. Wątpliwości te sformułował ostatecznie i jasno wyraził filozof Bar-Hillel w swoim raporcie z 1959 roku, w którym stwierdził, że stworzenie w pełni automatycznego systemu translacji maszynowej, który produkowałby przekłady o jakości porównywalnej z pracą zawodowego tłumacza w odniesieniu do języka ogólnego (ang. *general language*) jest po prostu niemożliwe i to nie tylko w chwili obecnej (przy obecnym stanie techniki), ale z zasady, z uwagi poważne ograniczenia o charakterze fundamentalnym [20]. Jako przykład, na którym oparł swoje rozumowanie była próba zautomatyzowania translacji następującej wypowiedzi:

Little John was looking for his toy train. Finally he found it. The train was in the pen. John was very happy.

Powstaje pytanie, jak podczas próby przekładu takiej wypowiedzi przetłumaczyć angielskie słowo „pen", bowiem słownik polsko-angielski oferuje w tym względzie co najmniej trzy podstawowe możliwości:

- Pen[1] – zagroda (w rozważanym przykładzie w znaczeniu zagroda z zabawkami lub raczej kącik z zabawkami)
- Pen[2] – przyrząd do pisania, długopis
- Pen[3] – łabędzica

Oczywiście człowiek tłumacz bez najmniejszego cienia zawahania opowie się za pierwszą z tych możliwości i przekład rozważanego zdania będzie wyglądał następująco:

Mały Jaś szukał swojej kolejki. W końcu ją znalazł. Kolejka znajdowała się w kąciku z zabawkami. Jaś był bardzo szczęśliwy.

Jednakże dokonanie takiego wyboru wymaga posiadania przez osobę tłumaczącą ogólnej wiedzy o świecie (trzeba wiedzieć, że kolejka może być w zagrodzie lub kąciku z zabawkami, a nie w długopisie, ani nie w łabędzicy). Oczywiście ludzie obdarzeni choć szczyptą inteligencji nie mają najmniejszych problemów z nabyciem takiej ogólnej wiedzy o świecie (ang. *general knowledge about the world*), jednakże jej wbudowanie w komputer może okazać się bardzo trudne lub wręcz niemożliwe.

Takie postawienie sprawy przez filozofa Bar-Hillela przyczyniło się w istotnym stopniu do zahamowania prac prowadzonych nad translacją automatyczną. Bezpośrednią konsekwencją wystąpienia Bar-Hillela było ogłoszenie w 1964 roku przez Automatic Language Processing Advisory Committee (ALPAC) wchodzącego w skład Amerykań-

skiej Akademii Nauk (ang. *US National Academy of Sciences*) raportu, nazwanego później Raportem ALPAC (ang. *ALPAC Report*), który potępił w czambuł wszelkie próby badań prowadzonych nad automatyzacją translacji. Raport ten stwierdzał, że, po pierwsze, nie ma żadnego dotkliwego braku zawodowych tłumaczy, a po drugie, że badania prowadzone nad translacją maszynową i tak do niczego sensownego nie doprowadzą, ponieważ konstrukcja systemu tłumaczącego ogólny język nie jest z zasady możliwa. W ślad za raportem ALPAC poszło obcięcie wszelkich funduszy rządowych wspierających badania prowadzone w tej dziedzinie, w związku z czym translacja automatyczna została odłożona do lamusa na dobrych kilkanaście lat.

Oczywiście argumenty zawarte w raporcie ALPAC są dyskusyjne. Pierwszy z nich, stwierdzający, że nie ma zapotrzebowania na tłumaczy jest wręcz absurdalny (było to może prawdą w Stanach Zjednoczonych w okresie Zimnej Wojny, gdzie zapotrzebowanie na tłumaczenie radzieckich dokumentów było całkowicie zaspokajane przez całe rzesze przeszkolonych w tym celu zawodowych tłumaczy – natomiast w żadnym wypadku nie jest to prawdą w odniesieniu do realiów europejskich, gdzie na stosunkowo niewielkim obszarze mamy do czynienia z prawdziwą mozaiką etniczną). Po drugie, z tego, że nie można zbudować systemu tłumaczącego język ogólny, charakteryzującego się jakością pracy porównywalną z pracą zawodowych tłumaczy, wcale nie wynika, że translacja maszynowa jest bezużyteczna. Poza tym, nawet w okresie powstania Raportu ALPAC na terenie Stanów Zjednoczonych i w Europie znajdowały się co najmniej trzy systemy automatycznej translacji, będące w powszechnym użytku. Znajdowały się one, między innymi, w bazie Sił Powietrznych Stanów Zjednoczonych (ang. *Wright Patterson USAF base*), w Amerykańskiej Komisji do Spraw Energii Atomowej (ang. *Oak Ridge Laboratory of the US Atomic Energy Commission*) oraz w Centrum do Spraw Energii Jądrowej EURATOM w miejscowości Ispra we Włoszech.

Po ogłoszeniu Raportu ALPAC szeregi badaczy zajmujących się translacja automatyczną znacznie stopniały, a ci co pozostali na swoich stanowiskach musieli szukać sponsorów w organizacjach pozarządowych, takich jak choćby Kościół Mormonów w stanie Utah, który był zainteresowany zautomatyzowaniem tłumaczenia tekstów biblijnych. Notabene, prowadzone w tym celu badania w Brigham Young University in Provo (Utah) doprowadziły do powstania dwóch komercyjnych systemów translacji maszynowej: WEIDNER i ALPS. Badania prowadzone były także w Kanadzie na Uniwersytecie w Montrealu (przez grupę TAUM), które doprowadziły do powstania komercyjnego systemu o nazwie METEO, którego zadaniem było tłumaczenie prognozy pogody z języka angielskiego na francuski. W Europie badania nad automatyczną translacją prowadziła, między innymi, grupa GETA w Grenoble oraz grupa SUSY w Saarbrücken. Interesujące badania prowadzone były również w byłym Związku Radzieckim, warto tutaj wspomnieć o grupach badawczych działających pod kierownictwem profesorów Mjelczuka i Apresjana [41].

Jednak prawdziwy renesans badania nad translacją maszynową przeżyły dopiero na początku lat osiemdziesiątych. Zaowocowały one pojawieniem się wielu komercyjnie dostępnych produktów, wśród których można wymienić system SYSTRAN, którego francusko-angielska wersja została zakupiona, między innymi, przez Komisję Wspólnot

Europejskich CEC (ang. *The Commission of the European Communities*). Również na Georgetown University w Waszyngtonie opracowano rosyjsko-angielski system, który został zakupiony przez Siły Powietrzne Stanów Zjednoczonych USAF i NASA. Na początek lat osiemdziesiątych przypada również prawdziwy boom systemów automatycznej translacji, jaki wystąpił w Japonii. Warto również wspomnieć o projektach finansowanych przez CEC, takich jak włosko-angielska wersja programu SYSTRAN oraz projekt EUROTA. W tym czasie powstał również finansowany przez Amerykańską Organizację Zdrowia PAHO (ang. *Pan American Health Organization*) system translacji maszynowej pomiędzy językiem hiszpańskim i angielskim o nazwie SPANAM. Z kolei na University of Texas w Austin został opracowany, przy finansowaniu przez Siły Powietrzne Stanów Zjednoczonych, system translacji maszynowej o nazwie METAL.

Tyle historii. O tym jak prezentuje się obecny stan badań nad translacją automatyczną można będzie przeczytać w kolejnym punkcie.

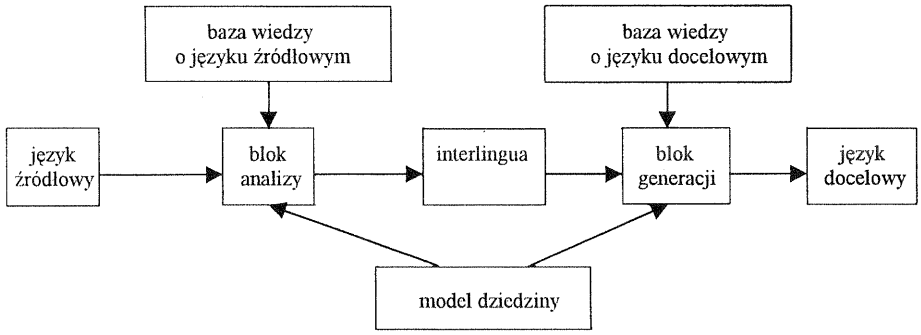
5. NOWOCZESNE SYSTEMY TRANSLACJI AUTOMATYCZNEJ

Obecnie badania nad translacją automatyczną prowadzone są w wielu ośrodkach badawczych na całym świecie. Poniżej omówione zostaną wybrane najnowsze osiągnięcia w tej dziedzinie.

5.1. SYSTEM KANT

Jako jeden z przykładów sytemu translacji automatycznej, którego konstrukcja zakończyła się pełnym sukcesem może posłużyć zrealizowany w Stanach Zjednoczonych na Carnegie-Mellon University system o nazwie KANT (ang. *Knowledge-based Accurate Natural-language Translation*). System KANT jest systemem automatycznej translacji, którego działanie oparte jest na wiedzy o języku (ang. *knowledge-based system*) i który zbudowany został w oparciu o architekturę wykorzystującą koncepcję interlingua. Schemat blokowy systemu KANT został zamieszczony na rys. 1 [21].

Koncepcja interlingua polega na przekształceniu, wykorzystując w tym celu bazę wiedzy o języku, wypowiedzi w języku źródłowym (czyli w tym, z którego chcemy tłumaczyć) do pewnej abstrakcyjnej reprezentacji – niezależnej od jakiegokolwiek rzeczywistego języka naturalnego – zwanej interlingua. Następnie wykorzystując bazę wiedzy o języku docelowym (czyli takim, na który chcemy tłumaczyć) z reprezentacji zapisanej w sztucznym języku *interlingua* generowana jest wypowiedź w języku docelowym. Zatem w procesie tłumaczenia wyodrębniają się wyraźnie dwa etapy. Pierwszy z nich, zwany etapem analizy, w którym z wypowiedzi w języku źródłowym generowany jest zapis wypowiedzi w języku *interlingua* oraz drugi, zwany etapem syntezy, w którym z zapisu w języku *interlingua* generowana jest wypowiedź w języku docelowym. W obu tych etapach (syntezy i analizy) wykorzystywana jest baza wiedzy o dziedzinie, której tłumaczone wypowiedzi dotyczą. Taka baza wiedzy stanowi model rozważanej dziedziny.



Rys. 1. Schemat architektury systemu translacji maszynowej KANT

System KANT został stworzony specjalnie pod kątem tłumaczenia dokumentacji technicznej i podręczników użytkowania sprzętu ciężkiego (koparki, dźwigi, spychacze itp.), który pewne przedsiębiorstwo amerykańskie eksportuje do licznych krajów europejskich. Zatem działanie sytemu KANT oparte jest na języku kontrolowanym, dotyczącym pewnej wąskiej dziedziny specjalistycznej. Obecnie odniesiono już duże rynkowe sukcesy tłumacząc z języka angielskiego przy użyciu sytemu KANT dokumentację techniczną sprzętu ciężkiego na języki takie jak: portugalski, hiszpański, francuski, włoski i niemiecki [22]. Należy przy tym zwrócić uwagę na fakt, że posiadając opracowany moduł analizy języka źródłowego systemu KANT, jego rozbudowa o nowy język polega jedynie na dołączeniu odpowiedniego bloku syntezy języka docelowego. Obecnie w Carnegie-Mellon University trwają prace nad opracowaniem takiego bloku syntezy dla języka tureckiego [23].

Również praca [24] stanowi ciekawy przykład adaptacji sytemu KANT do nowej dziedziny, gdzie w grę wchodziło tłumaczenie tekstów medycznych z języka angielskiego na włoski.

Zastosowanie koncepcji interlingua pozwala także na efektywne rozwiązanie problemu, który w systemach translacji automatycznej zwany jest problemem N^2 (N-kwadrat). Istota tego problemu jest następująca. Załóżmy, że mamy do czynienia z N językami i chcemy stworzyć system tłumaczący z każdego z tych języków na wszystkie pozostałe języki. Mamy zatem

$$C^N = \frac{N!}{2!\,(N-2)} = \frac{N(N-1)}{2} \tag{1}$$

par języków, pomiędzy którymi należy dokonać tłumaczeń. Ponieważ dla każdej pary języków konieczne jest stworzenie dwóch programów tłumaczących (z języka A na język B i z języka B na język A), liczba programów automatycznej translacji dla N języków wynosi

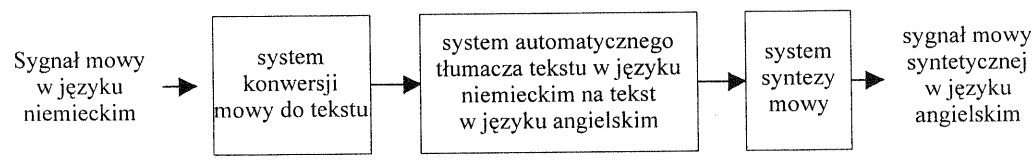
$$2C_2^N = N(N-1) \approx N^2 \tag{2}$$

Stąd pochodzi też nazwa problemu N^2. Natomiast w przypadku zastosowania koncepcji interlingua należy stworzyć jedynie N programów analizy (przejście z każdego z języków naturalnych do języka interlingua) oraz N programów syntezy (przejście z języka interlingua do każdego z języków naturalnych). Zatem w przypadku zastosowania koncepcji interlingua problem o złożoności kwadratowej zostaje zastąpiony problemem o złożoności liniowej [42].

Zatem rozwijanie systemów opartych o koncepcję interlingua wydaje się być bardzo obiecującym kierunkiem badań, jednakże nie ma jeszcze zgody co do tego, które ze stosowanych podejść w translacji maszynowej jest najlepsze. Możliwe, jak zresztą twierdzą autorzy pracy [25], że przyszłość translacji automatycznej należy do systemów hybrydowych, stanowiących połączenie różnych technik, takich jak: transfer struktur morfologicznych, koncepcja języka interlinga, tłumaczenie oparte na przykładach, metody statystyczne bazujące na niejawnych modelach Markowa, sztuczne sieci neuronowe itp.

5.2. PROJEKT VERBMOBIL

Obecnie badania nad translacją automatyczną prowadzone są również na Uniwersytecie Technicznym w Aachen w Republice Federalnej Niemiec. Opracowywany jest tam, miedzy innymi, projekt o nazwie Verbmobil, którego celem jest skonstruowanie systemu służącego do automatycznej translacji sygnału mowy (ang. *spontaneous speech translation*) wypowiadanej w języku niemieckim na sygnał mowy w języku angielskim [44]. System taki składa się z trzech głównych bloków, z których pierwszy stanowi system automatycznego rozpoznawania mowy i konwersji sygnału mowy do tekstu TTS (ang. *text-to-speech*). Z kolei drugi z bloków stanowi system automatycznego tłumacza (translacja z języka niemieckiego na język angielski), a ostatni z bloków spełnia rolę automatycznego syntezatora mowy i jego zadaniem jest generacja sztucznego sygnału mowy w języku angielskim, na podstawie tekstu otrzymanego z bloku automatycznego tłumacza. Schemat blokowy automatycznego translatora sygnału mowy zamieszczono na rys. 2.



Rys. 2. System automatycznego tłumacza sygnału mowy realizowany w ramach projektu Verbmobile

Do głównych cech projektu Verbmobile można zaliczyć:

- Konieczność uporania się z licznymi problemami jakie stwarza translacja z języka niemieckiego na język angielski. Trudności te są związane przede wszystkim z dość

dużymi różnicami w uporządkowaniu odpowiadających sobie słów w języku niemieckim i angielskim. Uwaga powyższa dotyczy zwłaszcza grup czasownikowych w obu językach. Ponadto w języku niemieckim występują liczne wyrazy będące zlepkiem wielu krótszych słów. W związku z tym wyrazy takie wymagają specjalnego potraktowania podczas translacji, gdyż muszą zostać zastąpione przez grupę kilku różnych słów języka angielskiego. Jako przykład posłużyć może zastąpienie niemieckiego słowa „Geschäftsreise" dwoma angielskimi słowami „business trip" (czyli podróż w celach służbowych).

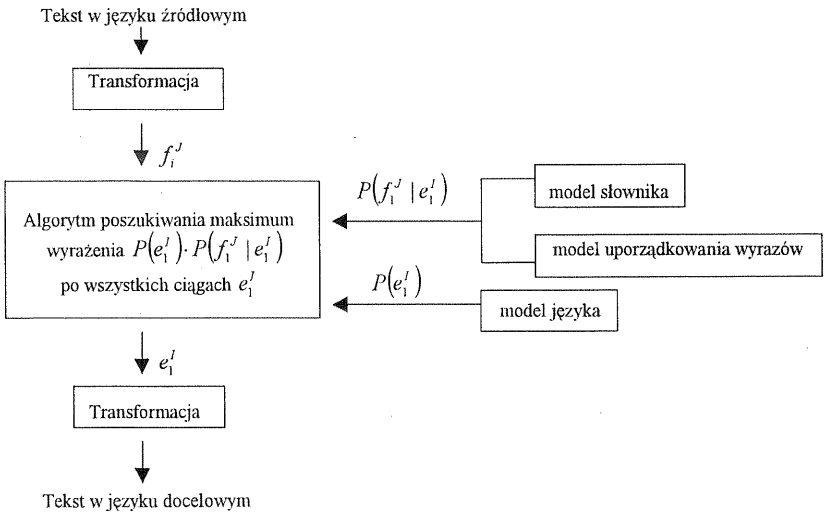
- W projekcie Verbmobile zastosowano trzy różne algorytmy statystyczne pozwalające na generacje sekwencji słów w języku angielskim, będących odpowiednikami zadanej wypowiedzi w języku niemieckim.
- Tłumaczeniu podlega sygnał mowy spontanicznej, w związku z czym występują tutaj dodatkowe problemy, w porównaniu np. z tłumaczeniem sygnału mowy będącej rezultatem odczytu tekstu pisanego. Przede wszystkim w mowie spontanicznej występuje dużo większa zmienność struktur syntaktycznych, ponadto generowane mogą być wypowiedzi nie do końca gramatycznie poprawne, będące np. rezultatem wahania się osoby mówiącej.
- W projekcie Verbmobile zastosowany został stosunkowo mały zbiór treningowy, składający się z nagranych wypowiedzi około 300 000 słów, należących do słownika o liczebności około 7 000 wyrazów.

5.2.1. Podejście statystyczne do zagadnienia automatycznej translacji

Celem automatycznej translacji jest dokonanie przekładu tekstu zapisanego w pewnym języku źródłowym (ang. *source language*) na język docelowy (ang. *target language*). Zatem dany jest pewien ciąg wyrazów w języku źródłowym (dla ustalenia uwagi niech będzie to np. język francuski), oznaczony jako $f_1^J = f_1 \ldots f_j \ldots f_J$. Zadaniem systemu jest dokonanie automatycznego przekładu tego ciągu na inny ciąg docelowy (dla ustalenia uwagi niech językiem docelowym będzie np. język angielski), oznaczony jako $e_1^I = e_1 \ldots e_i \ldots e_I$. Następnie spośród wszystkich możliwych ciągów docelowych wybierany jest ten, którego prawdopodobieństwo wystąpienia jest największe. Korzystając z reguły decyzyjnej Bayesa można zapisać:

$$e_1^I = \max_{e_1^I} \{P(e_1^I | f_1^J)\} = \max_{e_1^I} \{P(e_1^I) \cdot P(f_1^J | e_1^I)\} \tag{3}$$

W powyższym wzorze $P(e_1^I)$ jest tzw. modelem języka LM (ang. *Language Model*) dla języka docelowego, natomiast $P(f_1^J | e_1^I)$ jest modelem translacji. Z kolei operacja max oznacza poszukiwanie takiego ciągu wyrazów e_1^I w języku docelowym, dla którego iloczyn prawdopodobieństw ujętych w nawias klamrowy jest największy.
Architektura systemu automatycznego tłumacza została przedstawiona na rys. 3.



Rys. 3. Architektura systemu automatycznego tłumacza

Jak widać z rys. 3 system automatycznego tłumacza może zostać wzbogacony dodatkowo o bloki transformujące wejściowe i wyjściowe ciągi wyrazów. Transformacje te mogą dotyczyć operacji takich jak np. kategoryzacja pojedynczych wyrazów lub całych grup wyrazów bądź też bardziej zaawansowanych operacji przetwarzania wstępnego, które wymagają zastosowania odpowiedniego procesora tekstu (ang. *parsing*) [45].

5.2.2. Zagadnienie uporządkowania wyrazów w ciągach źródłowych i docelowych

Porównując wzajemne uporządkowanie wyrazów w ciągach stanowiących paralelne tłumaczenia tekstów, w przypadku języków indoeuropejskich zauważyć można silny efekt lokalizacyjny, polegający na tym, że danemu wyrazowi w ciągu źródłowym odpowiada w ciągu docelowym wyraz (bądź grupa wyrazów) zajmujących dokładnie taką samą lub bardzo zbliżoną pozycję. Zatem wzajemne przyporządkowanie wyrazów jest w większości przypadków monotoniczne. W celu matematycznego opisu tego zjawiska wprowadzane jest następujące odwzorowanie

$$j \rightarrow i = a_j \tag{4}$$

w którym wyrazowi f_j zajmującemu pozycję j w ciągu wejściowym, przypisywany jest wyraz e_i zajmujący pozycję $i = a_j$ w ciągu wyjściowym. Wprowadzenie rozważanego odwzorowania pozwala na następujący zapis modelu prawdopodobieństwa translacji

$$P(f_1^J \mid e_1^I) = P(J \mid I) \cdot \sum_{a_1^J} \prod_{j=1}^{J} [P(a_j \mid a_{j-1}, I, J) \cdot P(f_j \mid e_{aj})] \tag{5}$$

Powyższy wzór posiada trzy zasadnicze czynniki:

- Prawdopodobieństwo długości docelowego ciągu wyrazów $P(J|I)$
- Prawdopodobieństwo słownika translacji $P(f|e)$
- Prawdopodobieństwo uporządkowania wyrazów $P(a_j|a_{j-1}, I, J)$ – jest to tzw. model pierwszego rzędu

Ponadto, w celu redukcji liczby parametrów zakłada się, że wartość prawdopodobieństwa $P(a_j|a_{j-1}, I, J)$ zależy tylko od długości przeskoku $(a_j - a_{j-1})$ w uporządkowaniu w ciągu docelowym dwóch kolejnych wyrazów z ciągu źródłowego. Po wprowadzeniu oznaczeń $i = a_j$ oraz $i' = a_{j-1}$ otrzymuje się

$$P(i|i', I) = \frac{q(i-i')}{\sum_{i''=1}^{I} q(i''-i')} \tag{6}$$

gdzie $q(\cdot)$ oznacza pewną tablicę liczb nieujemnych, których wartości muszą zostać oszacowane na podstawie dwujęzycznego zbioru treningowego (paralelne tłumaczenia tekstów).

5.2.3. Trzy różne podejścia statystyczne do zagadnienia translacji

Celem zastosowanego w procesie translacji algorytmu optymalizacyjnego jest znalezienie najbardziej prawdopodobnej docelowej sekwencji wyrazów e_1^I o nieznanej z góry długości I, przy zadanej sekwencji źródłowej f_1^J o długości J.

W przypadku zastosowania modelu uporządkowania wyrazów pierwszego rzędu, otrzymuje się następujące kryterium optymalizacji

$$\max_I \left\{ P(J|I) \cdot \max_{e_1^I} \left\{ \prod_{i=1}^{I} P(e_i|e_j) \cdot \max_{a_1^I} P(f_1^J, a_1^J | e_1^I, J) \right\} \right\} \tag{7}$$

W pracy [1] zaproponowano trzy różne algorytmy poszukiwania najbardziej prawdopodobnej docelowej sekwencji wyrazów , które zostaną poniżej w skrócie omówione.

5.2.3.1. Kwazi-monotoniczne przyporządkowanie i poszukiwanie

W metodzie tej zakłada się, że wzajemne przyporządkowanie wyrazów pomiędzy ciągami źródłowym i docelowym spełnia warunek monotoniczności. Zatem dozwolone są jedynie przejścia z a_{j-1} do a_j na odległość δ, przy czym $\delta \equiv a_j - a_{j-1} \in \{0, 1, 2\}$. W związku z powyższym kryterium optymalizacyjne może zostać przepisane jako

$$\max_I \left\{ p(J|I) \cdot \max_{e_1^I a_1^J} \left\{ \prod_{j=1}^{J} [P(a_j|a_{j-1}, I) \cdot P_{[a_j, a_{j-1}]}(e_{a_j}|e_{a_{j-1}}) \cdot P(f_j|e_{a_j})] \right\} \right\} \tag{8}$$

W rozważanym przypadku złożoność obliczeniowa algorytmu poszukiwań wynosi $I_{\max} \cdot J \cdot E^2$, gdzie E oznacza rozmiar słownika języka docelowego, $I_{\max}$ a jest maksymalną długością docelowej sekwencji wyrazów, przyjmowaną jako $I_{\max} = 2 \cdot J$.

5.2.3.2. Odwrotne przyporządkowanie i poszukiwanie

W podejściu tym kluczowa idea polega na zorganizowaniu poszukiwań wzdłuż osi dopasowania pozycji wyrazów w ciągu docelowym. Wprowadza się w tym celu tzw. odwrotne przyporządkowanie (ang. *inverted alignment*) $i \rightarrow j = b_i$. W rozważanym przypadku kryterium optymalizacyjne może zostać zapisane w postaci

$$\max_{I} \left\{ P(J|I) \cdot \max_{e_1^I b_1^J} \left\{ \prod_{i=1}^{I} [P(e_i|e_{i-1}) \cdot P(i|j, I, J) \cdot P(f_i|e_j)] \right\} \right\} \tag{9}$$

Złożoność obliczeniowa rozważanego algorytmu poszukiwań wynosi $I_{\max}^2 \cdot J^2 \cdot E^2$.

5.2.3.3. Podejście oparte na wzorcach przyporządkowania

Głównym mankamentem rozważanych uprzednio modeli przyporządkowania pozycji wyrazów w ciągach źródłowym i docelowym jest fakt, iż odzwierciedlone są w nich zależności zachodzące jedynie pomiędzy pojedynczymi wyrazami. Natomiast w podejściu opartym na wzorcach przyporządkowania zamiast pojedynczych wyrazów rozważane są całe związki frazeologiczne. Zatem cała grupa sąsiadujących ze sobą wyrazów w ciągu źródłowym może zostać przyporządkowana innej grupie wyrazów w ciągu docelowym.

W celu dokonania matematycznego opisu podejścia opartego na wzorcach przyporządkowania, zarówno ciąg źródłowy f_1^J, jak i docelowy e_1^I przedstawione muszą zostać w postaci sekwencji grup wyrazów

$$f_1^J = \tilde{f}_1^K, \quad \tilde{f}_k = f_{j_{k-1}+1}, \ldots, f_{j_k}, \quad k = 1, \ldots, K \tag{10}$$

$$e_1^I = \tilde{e}_1^K, \quad \tilde{e}_k, \ldots, e_{i_k}, \quad k = 1, \ldots, K \tag{11}$$

W rozważanym przypadku model translacji opisany może zostać następującym wzorem

$$P(f_1^J|e_1^I) = P(\tilde{f}_1^K|\tilde{e}_1^K) = \sum_{\tilde{a}_1^K} \prod_{k=1}^{K} P(\tilde{a}_k|\tilde{a}_{k-1}, K) \cdot P(\tilde{f}_k|\tilde{e}_{\tilde{a}k}) \tag{12}$$

5.2.4. Ocena jakości algorytmów translacji

W pracy [44] zamieszczono wyniki testów pozwalających ocenić jakość przedstawionych powyżej trzech różnych statystycznych algorytmów translacji. Jednakże w celu przeprowadzenia takiej oceny potrzebny jest pewien rodzaj miary, która pozwoli określić liczbę błędów popełnianych podczas translacji. W pracy [44] zaproponowano trzy następujące kryteria błędów:

- Kryterium WER (ang. *Word Error Rate*). W przypadku tego kryterium liczba błędów jest wyliczana, jako minimalna liczba podstawień, wstawień i usunięć wyrazów, która musi zostać wykonana aby otrzymać wypowiedź poprawną pod względem syntaktycznym i semantycznym.
- Kryterium PWER (ang. *Position-independent Word Error Rate*). Kryterium to stanowi pewnego rodzaju modyfikację kryterium WER, którego główna zaletą jest to, że nie jest w nim wymagane perfekcyjnie dokładne uporządkowanie wyrazów – niezwykle trudne do osiągnięcia w przypadku systemów automatycznej translacji. W rozważanym kryterium porównywane są jedynie odpowiadające sobie słowa w ciągu źródłowym i docelowym, bez brania pod uwagę zajmowanych przez nie pozycji. Słowa, dla których nie zostały odnalezione odpowiedniki w ciągu źródłowym, traktowane są jako błędy podstawień. Z kolei w zależności od tego, czy ciąg docelowy jest dłuższy, czy krótszy od ciągu źródłowego, zbywające lub brakujące słowa traktowane są jako błędy wstawień bądź usunięć.
- Kryterium SSER (ang. *Subjective Sentence Error Rate*). Kryterium to opiera się na spostrzeżeniu, że w przypadku każdego systemu automatycznej translacji ostatecznym sędzią, dokonującym oceny jego jakości pracy jest korzystający z tego systemu człowiek – jest to zatem kryterium oceny subiektywnej. W kryterium tym przypisywany jest błąd o wartości zerowej uzyskanemu w procesie automatycznej translacji zdaniu, które jest poprawne zarówno pod względem syntaktycznym, jak i semantycznym. Z kolei zdaniu zbudowanemu nieprawidłowo pod względem syntaktycznym, ale mającemu poprawna semantykę przypisywany jest błąd równy 1/2 . Natomiast zdaniu nieprawidłowemu pod względem semantycznym przypisywany jest błąd równy jedności i to niezależnie od poprawności składniowej takiego zdania.

W przeprowadzonych podczas oceny projektu Verbmobile eksperymentach uzyskano procentowe stopy błędów translacji, które zamieszczone zostały w tab. 1.

Tabela 1

Uzyskane procentowe stopy błędów w przypadku zastosowania różnych algorytmów poszukiwań

Metoda	WER [%]	PER [%]	SSER [%]
Kwazi-monotoniczne przyporządkowanie i poszukiwanie	66,9	44,6	58,7
Odwrotne przyporządkowanie i poszukiwanie	64,4	50,9	61,5
Podejście oparte na wzorcach przyporządkowania	60,2	39,1	49,6

Na podstawie zamieszczonych w tab. 1 wyników widać, że działanie automatycznego systemu translacji sygnału mowy dalekie jest jeszcze od doskonałości. Powodem takiego stanu rzeczy jest fakt, że na liczbę popełnianych błędów posiada wpływ nie tylko automatyczny translator, ale również poprzedzający jego działanie system automatycznego rozpoznawania mowy. Niestety do chwili obecnej nie skonstruowano jeszcze systemu pozwalającego na stuprocentowo dokładne rozpoznanie sygnału mowy ciągłej należącej do słownika o dużych rozmiarach i to w sposób niezależny od mówcy [46].

Natomiast jeżeli chodzi o moduł automatycznego tłumacza, to jeszcze sporo wysiłku badawczego musi zostać włożone w rozwiązanie problemów takich jak optymalne grupowanie wyrazów w tłumaczone następnie wzorce oraz przetwarzanie wstępne polegające na odpowiednim dopasowaniu kolejności występowania wyrazów w języku źródłowym (ang. *word reordering*).

5.3. SYSTEM GBMT

Jednym z przodujących ośrodków badawczych, prowadzącym prace w dziedzinie translacji maszynowej jest amerykański ośrodek uniwersytecki *Carnegi-Mellon University* z siedzibą w Pittsburgu. W ośrodku tym opracowano metodę automatycznej translacji, bazującą na analizie uogólnionych przykładów GEBMT (ang. *Generalized Example-Bazed Machine Translation*) [50].

5.3.1. Czym jest translacja maszynowa oparta na przykładach?

Translacja maszynowa oparta na przykładach jest w swej istocie translacją poprzez analogię. Podczas przeprowadzania takiej translacji, systemowi zostaje zadany pewien zbiór zdań w języku źródłowym (ang. *source language*), z którego należy dokonać tłumaczenia, oraz zbiór odpowiadających im tłumaczeń na pewien język docelowy (ang. *target language*). Następnie system wykorzystuje zadane przykłady translacji, w celu dokonania automatycznego przekładu podobnych co do treści i formy tekstów zapisanych w języku źródłowym na język docelowy. Podstawowa przesłanka leżąca u podstaw konstrukcji takich systemów jest taka, że jeżeli uprzednio poprawnie przetłumaczony fragment tekstu pojawi się jeszcze raz, wówczas odpowiadający mu przekład w języku docelowym będzie również i w tym przypadku poprawny.

W zależności od wyboru konkretnej metody, na której opiera się działanie takiego systemu, stosowane są różne sposoby ustalenia, które z przykładów translacji uznaje się za identyczne bądź dostatecznie podobne do fragmentów dokumentu podlegającego właśnie procesowi translacji.

Pewna ograniczona forma systemu translacji opartej na przykładach jest już od dawna dostępna komercyjnie i znana pod nazwą pamięci translacyjnej (ang. *translation memory*). W przypadku pamięci translacyjnej, podczas tłumaczenia tekstu przez użytkownika, dokonane przez niego przekłady są systematycznie dołączane do bazy danych.

W przypadku, gdy takie samo zdanie pojawi się ponownie, uprzednio dokonane tłumaczenie zostaje włączone do tłumaczonego dokumentu. Postępowanie takie zaoszczędza użytkownikowi wysiłku, polegającego na ponownym tłumaczeniu takiego samego fragmentu tekstu. Rozważany sposób translacji okazuje się być niezwykle efektywny, w przypadku dokonywania przekładu kolejnej wersji uprzednio przetłumaczonego dokumentu – zwłaszcza, gdy naniesione poprawki mają niewielki zakres.

5.3.2. Na czym polega uogólnienie przykładów?

Największy problem podczas praktycznej realizacji systemów automatycznej translacji opartej na uogólnionych przykładach polega na tym, że systemy takie potrzebują ogromnej ilości danych, w postaci uprzednio przetłumaczonych (przez tłumaczy profesjonalistów) tekstów. Aby uzyskać system, charakteryzujący się przyzwoitą jakością tłumaczonych tekstów, należy zwykle zadać przykłady translacji w postaci tekstów o objętości co najmniej kilku milionów słów. Zatem poszukiwane są metody jakiegoś uogólnienia zebranych w bazie danych przykładów, tak aby ich mniejsza liczba wystarczała już do dokonania poprawnych tłumaczeń.

Rozważony zostanie następujący przykład translacji zadania z języka angielskiego na język niemiecki:

John Hancock was in Philadelphia on July 4th. (j. angielski)
John Hancock war am 4. Juli in Philadelphia. (j. niemiecki)

Gdyby system posiadał odpowiednie wiadomości na temat tego, że John Hancock jest osobą <PERSON>, Philadelphia jest miastem <CITY>, a July 4th jest datą <DATE>, przykład taki mógłby być zapisany w bazie danych jako:

<PERSON> was in <CITY> on <DATE>. (j. angielski)
<PERSON> war am <DATE> in <CITY>. (j. niemiecki)

W rozważanym przypadku wyrazy <PERSON>, <CITY> i <DATE> pełnią funkcję specjalnych żetonów (ang. *tokens*), służących do oznaczania klas równoważności, których poszczególne elementy mogą być dowolnie podstawiane w miejsce żetonu. Należy zwrócić uwagę, że obecnie system jest w stanie wygenerować automatyczne tłumaczenia wielu takich analogicznych ciągów wyrazów, bez konieczności ograniczania się do człowieka o imieniu John Hancock, Filadelfii i dnia czwartego lipca. Bowiem dowolny członek klasy równoważności typu <PERSON> może zostać podstawiony w miejsce zajmowane oryginalnie przez grupę wyrazów John Hancock.

Powstaje jednakże naturalne pytanie, skąd system ma wiedzieć, że John Hancock jest osobą? Otóż dokonane to musi zostać poprzez wprowadzenie odpowiednich atrybutów w bazie danych. Podczas tworzenia bazy danych niektóre z rekordów muszą zostać opatrzone specjalnymi etykietkami typu <PERSON>, <CITY> bądź <DATE>. Podczas tłumaczenia, słowa takie zostają zastąpione odpowiednimi żetonami i dopiero wtedy poszukuje się w bazie danych odpowiednich wzorców translacji.

Ponadto poszczególni członkowie klasy równoważności mogą sami zawierać żetony, co pozwala na zdefiniowanie wzorca. Na przykład tłumaczenie wzorca DATE może zostać zdefiniowane następująco:

<MONTH> <NUMBER>1, <NUMBER>2 (j. angielski)
<NUMBER>1. <MONTH> <NUMBER>2 (j. niemiecki)

Zatem w oparciu o taki uogólniony i zagnieżdżony wzorzec otrzymamy następujące tłumaczenie

July 4, 1776 (j. angielski)
4. Juli 1776 (j. niemiecki)

Innym sposobem uogólnienia wzorców jest wprowadzenie w bazie danych specjalnych etykietowanych rekordów (ang. *tagged entries*). Rekordy takie różnią się od rekordów zawierających żetony tym, że mogą posiadać wiele znaczeń i nie są bezpośrednio stosowane do każdego wejściowego ciągu wyrazów, ale jedynie w przypadku gdy obecny jest pewien kontekst wypowiedzi. Ponieważ etykietowane rekordy są przechowywane w tej samej bazie danych, co i rekordy zwykłe, mogą one brać udział w poszukiwaniu wzorca tłumaczenia zarówno przed jak i po zastąpieniu wybranych słów żetonami. Również etykietowane rekordy mogą same w sobie zawierać żetony. Zatem konieczne jest wprowadzenie procedury rekurencyjnej. Mianowicie wybrane słowa zastępowane są żetonami, a z kolei żetony te mogą zostać zastąpione innymi żetonami. Procedura taka kontynuowana jest do momentu, aż żadne kolejne podstawienie nie jest już możliwe.

5.3.3. Efektywność uogólnionych przykładów

W celu oceny stopnia efektywności podejścia opartego na uogólnionych przykładach. Przeprowadzono eksperymenty porównujące skuteczność działania trzech różnych systemów, opartych na bazach danych o odmiennej organizacji:

1. Baza danych (1) bez żetonów (z wyjątkiem tych zastępujących liczby).
2. Baza danych (2) wykorzystująca jedynie żetony pojedynczego poziomu w celu zastąpienia pewnych wyrazów.
3. Baza danych (3) wykorzystująca żetony i w pełni rekurencyjne zastępowanie żetonami wyrazów, jak również zastosowanie etykietowanych rekordów.

Badany był system automatycznego tłumaczenia z języka francuskiego na język angielski. W bazie danych zawarto 76000 wzorców struktur morfologicznych oraz 550 reguł gramatycznych. Wyniki eksperymentów automatycznej translacji zawarto w tab. 2. W tabeli tej podano w procentach wielkość pokrycia tłumaczonego tekstu uogólnionymi przykładami translacji pochodzącymi z różnych typów baz danych.

Jednakże sama procentowa wartość pokrycia uogólnionymi przykładami translacji tekstu wejściowego nie stanowi jedynej miary jakości pracy systemu. Bowiem, kolej-

Tabela 2

Wyniki eksperymentów automatycznej translacji dla trzech różnych baz danych

Ilość słów w bazie danych	Baza danych (1)	Baza danych (2)	Baza danych (3)
1.000.000	78%	80%	94%
2.000.000	82%	84%	95%
3.000.000	85%	86%	96%
4.000.000	88%	89%	96,5%
5.000.000	89%	90%	97%

nym ważnym czynnikiem jest średni rozmiar fragmentów tekstu, dla których system dysponuje przykładami translacji. W ogólności, zastosowanie przykładów translacji o większej długości (większej liczbie wyrazów) daje zdecydowanie lepsze rezultaty (wyższa jakość tłumaczenia), ponieważ dłuższe przykłady zawierają więcej kontekstu i stąd prawdopodobieństwo przypisania grupie wyrazów niewłaściwego znaczenia staje się zdecydowanie mniejsze.

W tab. 3 zamieszczono wyniki eksperymentów automatycznej translacji z języka hiszpańskiego na język angielski, tym razem jednak podano średnią długość (wyrażoną w liczbie wyrazów) odnajdywanych w bazie danych przykładów tłumaczeń.

Tabela 3

Średnia długość odnajdywanych w bazach danych przykładów tłumaczeń

Ilość słów w bazie danych	Baza danych (1)	Baza danych (2)	Baza danych (3)
2.000.000	3,5	3,6	3,9
4.000.000	3,7	3,8	4,1
6.000.000	3,8	3,9	4,2
8.000.000	3,9	4,1	4,3
10.000.000	4,1	4,2	4,4

Jak widać z przedstawionych tabeli samo wprowadzenie żetonów poprawia wyniki zaledwie o kilka procent – dotyczy to zarówno wielkości pokrycia uogólnionymi przykładami tłumaczonego tekstu, jak i średniej długości tłumaczonych przykładów. Natomiast wprowadzenie przykładów zagnieżdżanych rekurencyjnie powoduje znaczne poprawienie wyników.

Jednakże z punktu widzenia zastosowań praktycznych, bardziej istotną miarą jakości pracy systemu jest to, jak duża porcja tekstu z paralelnym tłumaczeniem jest wymagana, aby system był w stanie tłumaczyć (w miarę poprawnie) dowolne teksty. W tym miejscu wprowadzenie rekurencyjnie zagnieżdżonych przykładów wprowadza fundamentalną zmianę. Na przykład w przypadku dokonywania przekładów z języka francuskiego na

język angielski, w przypadku wprowadzenia jedynie żetonów, do pokrycia 80% wejściowego tekstu potrzeba bazy danych z paralelnymi tekstami o łącznej objętości około 1,4 miliona słów. Natomiast wprowadzenie pełnej metody rekurencyjnego dopasowywania uogólnionych przykładów redukuje tę liczbę do zaledwie 280 tysięcy. W przypadku 90% pokrycia tłumaczonego tekstu różnica ta staje się jeszcze większa, ponieważ dla systemu stosującego jedynie żetony bez rekurencji rozmiar bazy danych musi być aż 11 razy większy.

5.3.4. Przykłady zastosowań

Opracowywane w Carnegi-Mellon University techniki automatycznej translacji oparte na uogólnionych przykładach służą nie tylko do dokonywania w trybie off-line tłumaczenia tekstów, ale także znajdują zastosowanie w wielu różnorodnych aplikacjach. Przykładami takich aplikacji może być tłumaczenie w czasie rzeczywistym sygnału mowy oraz wyszukiwanie informacji w obcojęzycznych dokumentach.

Jako przykład systemu do automatycznej translacji sygnału mowy może posłużyć rozwijany w Carnegi-Mellon University program o nazwie DIPLOMAT. Obecna wersja tego programu pozwala na dwustronne tłumaczenie sygnału mowy, pomiędzy językiem angielskim, a językami takimi, jak: chorwacki, kreolski, koreański oraz hiszpański. W projekcie DIPLOMAT jako system rozpoznawania mowy ciągłej zastosowano opracowany wcześniej system Sphinx-II, którego celem jest dokonywanie automatycznej konwersji wypowiedzi użytkownika do postaci tekstowej. Otrzymany w ten sposób tekst po dokonaniu automatycznej korekty (opartej na modelu języka) poddawany jest tłumaczeniu maszynowemu opartemu na uogólnionych przykładach. Następnie przetłumaczony tekst podawany jest na wejście syntezatora mowy o nazwie Phonebox – który jest kolejnym z projektów opracowanych w *Carnegi-Mellon University*.

Ponieważ projekt DIPLOMAT jest dwukierunkowym systemem automatycznej translacji sygnału mowy wykorzystuje on dwa różne zestawy oprogramowania, z których pierwszy służy np. tłumaczeniu języka angielskiego na język hiszpański, a drugi służy do tłumaczenia w druga stronę, tzn. z języka hiszpańskiego na język angielski. Rozwiązanie takie pozwala nie tylko na tłumaczenie w obie strony, ale także na dokonanie podwójnego tłumaczenia z języka pierwszego na drugi i ponownie z języka drugiego na pierwszy. Jeżeli takie podwójne tłumaczenie nie zmienia w istotny sposób sensu wypowiedzi, wówczas jest dobrym gwarantem poprawności pracy systemu.

Opracowywane są również i inne systemy służące do automatycznego tłumaczenia w czasie rzeczywistym języka mówionego. Przykładem takiego systemu jest np. NICE, którego zadaniem jest opracowanie automatycznych programów tłumaczących pomiędzy językiem angielskim, a językami mniejszości narodowych zamieszkujących Stany Zjednoczone. Z kolei projekt JANUS został skoncentrowany na dziedzinie dotyczącej podróży i turystyki, a jego zadaniem jest umożliwienie konwersacji pomiędzy pracownikami biur podróży, agencji turystycznych, lotnisk itp. a klientami, którzy nie znają języka angielskiego. Obecnie uczestnicy programu JANUS pracują nad takimi językami jak angielski, niemiecki, japoński, koreański, włoski i francuski.

Drugą dziedziną zastosowań technologii języka naturalnego, której znaczenie ostatnio bardzo wzrosło jest wyszukiwanie informacji w obcojęzycznych dokumentach (ang. *cross-language information retrieval*). W przypadku takich systemów jedynym rozsądnym podejściem jest dokonanie tłumaczenia zapytania użytkownika, ponieważ tłumaczenie wszystkich dokumentów, które potencjalnie mogłyby zawierać potrzebne informacje nie wchodzi raczej w rachubę. Jednakże zapytania użytkownika zawierają często izolowane słowa bądź krótkie frazy (rzadziej pełne zdania), które w związku z tym mogą sprawiać pewne kłopoty programowi maszynowego tłumacza, z uwagi na ich możliwą wieloznaczność.

W praktyce, systemy automatycznej translacji współpracują często również z modułami statystycznego modelowania języka. Mówiąc w wielkim skrócie, statystyczne modele języka traktują wypowiedzi w języku naturalnym jako ciąg pewnych arbitralnie zadanych symboli, a ich zadaniem jest udzielenie odpowiedzi na pytanie, jak duże jest prawdopodobieństwo, tego że zadana wypowiedź należy do danego języka oraz posiada poprawną budowę gramatyczną. W kolejnym punkcie zostanie przybliżony czytelnikowi aktualny stan badań nad statystycznym modelowaniem języka, przy czym szczególna uwaga zostanie skoncentrowana na nowych trendach pojawiających się w tej dziedzinie.

6. STATYSTYCZNE MODELOWANIE JĘZYKA

Statystyczne modelowanie języka SLM (ang. *Statistical Language Modeling*) stanowi próbę uchwycenia regularności, jakie występują w każdym z języków naturalnych. Celem tego zabiegu jest poprawienie jakości działania różnorodnych aplikacji komputerowych, które wykorzystują podczas swego działania język naturalny [48]. Mówiąc ogólnie, głównym celem statystycznego modelowania języka jest aproksymacja funkcji dystrybucji prawdopodobieństwa, w odniesieniu do różnorodnych jednostek lingwistycznych, takich jak np. pojedyncze słowa, zdania czy też całe dokumenty.

Statystyczne modelowanie języka odgrywa ważną rolę w wielu obszarach zastosowań, do których można zaliczyć rozpoznawanie mowy, systemy automatycznego tłumaczenia (ang. *machine translation*), klasyfikację dokumentów, rozpoznawanie pisma, automatyczną korektę pisowni (ang. *spelling correction*) i wiele innych.

6.1. PODSTAWOWE DEFINICJE

Statystyczny model języka jest dystrybucją prawdopodobieństwa $P(s)$ rozciągniętą na wszystkie możliwe jednostki leksykalne s. Statystyczne modelowanie języka jest często porównywane z tzw. lingwistyką obliczeniową (ang. *computational linguistics*), ponieważ obie wymienione dziedziny wiedzy posiadają dość rozmyte granice i wiele rozpatrywanych zagadnień występuje w obu z nich. Jednakże występują pomiędzy nimi także dość poważne różnice, z których najważniejsza może zostać opisana w sposób następujący. Niech S będzie sekwencją wyrazów pewnej zadanej wypowiedzi oraz nich

H będzie pewną ukrytą strukturą zawartą w tej wypowiedzi, np. H może być sensem tej wypowiedzi. Zatem w przypadku statystycznego modelowania języka zainteresowanie koncentrowane jest głównie wokół zagadnień wyznaczenia dystrybucji prawdopodobieństwa P(S), podczas gdy w przypadku lingwistyki obliczeniowej celem jest znalezienie dystrybucji prawdopodobieństwa P(H|S). Oczywiście jeżeli znane jest prawdopodobieństwo łączne P(S,H), można z niego wyprowadzić zarówno P(S), jak i P(H|S). Niestety w praktyce okazuje się, że wyznaczenie takiego łącznego prawdopodobieństwa nie jest możliwe [47].

Statystyczne modelowanie języka znajduje zastosowanie w systemach rozpoznawania mowy, zwłaszcza w kontekście zastosowań tzw. klasyfikatorów Bayesa, gdzie modele takie spełniają funkcje modułów wyznaczających tzw. prawdopodobieństwo *a priori*. Na przykład w systemie automatycznego rozpoznawania sygnału mowy, przy zadanym sygnale akustycznym a, celem jest znalezienie sekwencji słów s, której prawdopodobieństwo wypowiedzenia jest największe. W kontekście zastosowania klasyfikatora Bayesa problem ten wygląda następująco:

$$s^* = \arg \max_s P(s|a) = \arg \max_s P(a|s)\,P(s) \tag{13}$$

We wzorze (13) model języka P(s) odgrywa rolę prawdopodobieństwa a priori.

Z kolei w przypadku zagadnienia automatycznej klasyfikacji dokumentów, przy zadanym dokumencie d celem jest znalezienie klasy c, do której dokument ten należy. W takim przypadku podane są zwykle przykłady dokumentów z każdej z klasy i dla każdej klasy utworzony jest odrębny model języka $\{P_1(d), P_2(d), \ldots, P_k(d)\}$. Po zastosowaniu klasyfikatora Bayesa rozwiązanie wygląda następująco

$$c^* = \arg \max_c P(c|d) = \arg \max_c P(d|c)\,P(c) \tag{14}$$

We wzorze (14) model języka $P(c|d)$ spełnia rolę miernika prawdopodobieństwa przynależności zadanego dokumentu do odpowiedniej klasy.

6.2. OCENA JAKOŚCI MODELU

Miarą jakości każdej techniki modelowania języka jest wielkość prawdopodobieństwa pojawienia się nowych (nie znanych uprzednio przez ten model danych). Prawdopodobieństwo to można wyrazić wzorem

$$P(D|M) = \frac{1}{n} \sum_i^n \log P_M(D_i) \tag{15}$$

We wzorze tym $D = (D_1, D_2, \ldots, D_n)$ oznacza nową próbkę danych, natomiast M jest modelem języka M. Niekiedy wyznacza się również entropię *Ent* próbki D, daną wzorem

$$Ent = -\sum_{D} P(D) \log P_M(D) \tag{16}$$

Często używanym terminem jest również złożoność modelu (ang. *perplexity*) definiowana jako

$$Per = 2^{Ent} \tag{17}$$

Jednakże ostatecznie jakość skonstruowanego modelu języka musi być mierzona z punktu widzenia aplikacji, dla której model ten został utworzony. W tym przypadku niezawodnym miernikiem jakości jest ilość błędów popełnianych przez rozważaną aplikację. Niestety funkcje opisujące częstość pojawiania się błędów są zwykle nieliniowe, a ich zależność od dobranych parametrów modeli nie jest zwykle znana. Z obserwacji wynika, że zwykle mniejsza wartość współczynnika *Per* daje w rezultacie niższą stopę błędów, ale nie jest to bezwzględną regułą. Zwykle redukcja współczynnika *Per* o 5% nie jest w ogóle zauważalna z punktu widzenia jakości funkcjonowania danej aplikacji. Redukcja tego współczynnika o 10%–20% jest zwykle godna uwagi i zazwyczaj powoduje pewną widoczną poprawę w funkcjonowaniu aplikacji. Niestety w praktyce rzadko udaje się uzyskać redukcję o 30% i więcej, co daje już bardzo znaczącą poprawę.

6.3. SŁABE STRONY DOTYCHCZASOWYCH MODELI JĘZYKA

Zastosowanie nawet najprostszego modelu języka posiada bardzo duży wpływ na sposób funkcjonowania aplikacji. Efekt ten może zostać najłatwiej zaobserwowany przez usunięcie modelu języka z dowolnego systemu rozpoznawania mowy. Jednakże stosowane obecnie techniki modelowania języka są jeszcze dalekie od optymalnych. Taki stan rzeczy ma kilka istotnych powodów, do których najczęściej zalicza się:

- Zależność modelu od dziedziny, której język dotyczy. Obecnie stosowane modele języka są bardzo wrażliwe na zmiany w stylu, temacie i rodzaju tekstu, w oparciu o który zostały wytrenowane. Na przykład, w celu modelowania typowej konwersacji telefonicznej osiąga się znacznie lepsze rezultaty używając zbioru treningowego zawierającego 2 miliony słów transkrybowanych z typowych rozmów telefonicznych, niż 140 milionów słów transkrybowanych z wiadomości telewizyjnych i radiowych. Efekt ten okazuje się bardzo silny nawet w przypadku zmian, które z punktu widzenia człowieka wydają się być nieistotne. Na przykład w przypadku systemu modelowania języka opartego na tekstach zaczerpniętych z agencji informacyjnej Dow-Jones Newswire, liczba popełnianych błędów wzrosła dwukrotnie w przypadku zastosowania tego systemu do analizy tekstów zaczerpniętych z agencji informacyjnej Associated Press (pochodzących dodatkowo z tego samego okresu czasu).

- Fałszywe założenie o niezależności poszczególnych elementów języka. Celem możliwości implementacji w systemie komputerowym, wszystkie istniejące techniki modelowania języka zakładają pewną formę niezależności pomiędzy różnymi częściami tego samego dokumentu. Na przykład w przypadku najczęściej stosowanych modeli języka, bazujących na tzw. n-gramach, zakłada się, że prawdopodobieństwo wystąpienia kolejnego słowa w danej sekwencji słów zależy jedynie od obecności n-1 słów poprzednich. Nawet pobieżny rzut oka na dowolny tekst zapisany w języku naturalnym przekonuje o fałszywości takiej przesłanki.
- Brak wprzęgnięcia ludzkiej wiedzy o języku do istniejących modeli. Wiedza, jaką każdy człowiek posiada o używanym przez niego języku mogłaby okazać się bardzo użyteczna w celu ograniczenia entropii istniejących modeli języków. Eksperymenty takie przeprowadził między innymi Shannon w latach osiemdziesiątych, a ich celem było ograniczenie entropii modeli języka angielskiego. Eksperymenty te polegały na tym, że wybranym osobom podawano pewną próbkę tekstu, celem wskazania kolejnego elementu, jaki powinien po zadanym fragmencie wystąpić. Przeprowadzone eksperymenty dowiodły w sposób jednoznaczny, że człowiek kierując się zdrowym rozsądkiem, wnioskowaniem lingwistycznym i znajomością dziedziny, której dotyczy dany model języka, może (wykonując rutynowe czynności) przyczynić się do znacznego wzrostu jakości modelu języka.

6.4. PRZEGLĄD NAJNOWSZYCH TECHNIK I TRENDÓW W STATYSTYCZNYM MODELOWANIU JĘZYKA

Praktycznie we wszystkich modelach języka naturalnego stosuje się dekompozycję prawdopodobieństwa wystąpienia sekwencji wyrazów na iloczyn odpowiednich prawdopodobieństw

$$P(s) = P(w_1 \ldots w_n) = \prod_{i=1}^{n} P(w_i | h_i) \tag{18}$$

gdzie w_i jest i-tym słowem w sekwencji wyrazów s, natomiast $h_i = \{w_1, w_2, \ldots, w_{i-1}\}$ jest nazywane historią.

6.4.1. Modele oparte na n-gramach

Najczęściej wykorzystywaną obecnie technikę modelowania języka stanowią tzw. n-gramy. Są one wykorzystywane praktycznie w każdym komercyjnym systemie rozpoznawania mowy. Technika n-gramów polega na estymacji prawdopodobieństwa wystąpienia zadanej sekwencji wyrazów przy użyciu łańcuchów Markowa rzędu n-1.

$$P(w_i | h_i) \approx P(w_i | w_{i-n+1}, \ldots, w_{i-1}) \tag{19}$$

W praktyce najczęściej wykorzystywane są tzw. trigramy (n = 3), w przypadku dysponowania odpowiednio dużym zbiorem treningowym oraz tzw. bigramy (n = 2), dla zbiorów treningowych o mniejszych rozmiarach.

Jednakże dokładne wyznaczenie prawdopodobieństw wszystkich trigramów, a nawet bigramów jest wciąż jeszcze nie do końca rozwiązanym problemem. Powodem tego jest po prostu zbyt mała liczba danych jakie można pozyskać z istniejących dla danego języka dokumentów. Na przykład w przypadku wypisania wszystkich trigramów, jakie można znaleźć w artykułach wziętych z codziennych gazet o łącznej objętości 38 milionów słów, okazuje się, że w nowo wziętym artykule co trzeci trigram jest nieznany. Ponadto nawet w przypadku zaobserwowanych trigramów zdecydowana większość z nich pojawiła się tylko raz. Zatem ocena prawdopodobieństwa wystąpienia danego trigramu nie jest sprawą łatwą – głównie ze względu na szczupłość danych treningowych.

Jednym ze sposobów pozwalających na zaradzenie takiej niekorzystnej sytuacji jest grupowanie słownictwa w szersze klasy. Można to zrobić na kilka różnych sposobów, przy czym do najczęściej stosowanych można zaliczyć:

$$P(w_3|w_1, w_2) = P(w_3|C_3)P(C_3|w_1, w_2) \tag{20}$$

$$P(w_3|w_1, w_2) = P(w_3|C_3)P(C_3|w_1, C_2) \tag{21}$$

$$P(w_3|w_1, w_2) = P(w_3|C_3)P(C_3|C_1, C_2) \tag{22}$$

$$P(w_3|w_1, w_2) = P(w_3|C_1, C_2) \tag{23}$$

We wzorach powyższych symbol C_i oznacza klasę, do której słowo w_i zostało przypisane. Jakość powstałych w ten sposób modeli zależy oczywiście od obranego sposobu grupowania słów w klasy. W przypadku zastosowań dotyczących pewnych wąskich dziedzin stosunkowo dobre rezultaty daje manualne grupowanie słów w kategorie semantyczne. Niestety w przypadku zastosowań dotyczących szerszych dziedzin taki sposób postępowania nie przynosi zwykle pożądanych rezultatów.

W literaturze przedmiotu znane są przykłady systemów, w których po zastosowaniu automatycznego grupowania wyrazów uzyskano redukcję stopnia złożoności modelu języka o około 10%, co dało już zauważalne rezultaty w przypadku aplikacji z dziedziny rozpoznawania sygnału mowy.

6.4.2. Modele oparte na drzewach decyzyjnych

W modelowaniu języka drzewa decyzyjne wprowadza się celem dokonania arbitralnego podziału przestrzeni historii h ciągu wyrazów, poprzez zadawanie w każdym z węzłów drzewa binarnych pytań dotyczących tej historii. Dane treningowe wykorzystywane są w tym przypadku do konstrukcji dystrybucji prawdopodobieństwa $P(w|h)$ pojawienia się następnego słowa.

Zastosowanie techniki drzew decyzyjnych do modelowania języka stanowi bardzo duże wyzwanie. Przestrzeń historii h jest bowiem bardzo duża (rzędu 10^{100} dla

20-wyrazowej sekwencji z 100000 słownika). Jeszcze większe rozmiary ma przestrzeń możliwych pytań $2^{10^{100}}$. Jeżeli nawet pytania takie zostaną ograniczone do poszczególnych słów występujących w historii h, to i tak ich liczba wyniesie $20 \cdot 2^{10^5}$. W praktyce w celu ograniczenia rozmiarów przestrzeni poszukiwań stosuje się tak zwane algorytmy zachłanne (ang. *greedy search algorithms*).

Niestety uzyskiwane w ten sposób rezultaty nie są najlepsze. Przykładem tego może być system CART, w którym zastosowano historię o długości 20 słów oraz ograniczono pytania do poszczególnych słów. Trening rozważanego systemu zajął kilka miesięcy, a uzyskanym rezultatem była redukcja stopnia złożoności modelu *Per* zaledwie o 4%. Na zakończenie autorzy tego projektu stwierdzili, że jest rzeczą możliwą, iż istnieją drzewa decyzyjne, które znacznie przewyższają jakością działania n-gramy. Niestety znalezienie ich w chwili obecnej nie wydaje się być możliwe, zarówno z powodów obliczeniowych, jak i z braku zbiorów treningowych o odpowiednich rozmiarach.

6.4.3. Modele bazujące na motywacji lingwistycznej

W pewnym sensie motywacją leżącą u podstaw każdego z modeli języków była przynajmniej pewna intuicyjna wiedza o języku. Niestety w większości aktualnie stosowanych modeli języka nastąpiło odejście od języka, bowiem w modelach tych poszczególne jednostki lingwistyczne traktowane są jako ciągi arbitralnie zadanych symboli, w szczególności mogą one nie reprezentować w ogóle żadnego języka. Zatem dalszą poprawę jakości działania istniejących modeli można uzyskać jedynie na drodze ponownego wprowadzenia wiedzy o języku do tych modeli.

Do dwóch najczęściej wykorzystywanych w tym celu technik należą tzw. gramatyki bezkontekstowe CFG (ang. *Context-Free Grammmars*) oraz gramatyki połączeń (ang. *Link Grammars*).

Gramatyka bezkontekstowa definiowana jest przez słownictwo, zbiór symboli nieterminalnych oraz zbiór reguł przejść. Wypowiedzi są generowane po wystartowaniu z początkowego symbolu nieterminalnego i sekwencyjne stosowanie reguł przejść. Każda z takich reguł przejść przekształca jeden symbol nieterminalny w sekwencję słów i symboli nieterminalnych. Operacja taka jest stosowana do momentu kiedy wygenerowane zostaną jedynie słowa stanowiące symbole terminalne.

Z kolei gramatyka połączeń stanowi konstrukcję, w której każde ze słów połączone jest z jednym lub większą liczbą zbiorów łączników. Wypowiadana sekwencja wyrazów jest przedstawiana zatem w postaci pewnego grafu planarnego.

6.4.4. Modele eksponencjalne

Głównym problemem wszystkich przedyskutowanych do tej pory modeli języka jest brak odpowiednich danych treningowych. Jest to zwłaszcza widoczne w przypadku drzew decyzyjnych, gdzie w miarę rozrostu drzewa kolejne jego gałęzie zawierają coraz

mniej danych treningowych. W celu przezwyciężenia tych trudności zaproponowano modele eksponencjalne. Model eksponencjalny dany jest wzorem

$$P(w|h) = \frac{1}{Z(h)} \exp\left[\sum_i \lambda_i f_i(h, w)\right] \tag{24}$$

gdzie
λ_i – parametry modelu
$Z(h)$ – czynnik normalizacyjny
$f_i(h, w)$ – arbitralnie zadane funkcje par (historia, słowo)
W literaturze przedmiotu znaleźć można przykłady systemu rozpoznawania mowy, w którym po zastosowaniu eksponencjalnego modelu języka stopa popełnianych błędów uległa redukcji o 14% w stosunku do zastosowanych uprzednio trigramów. Jednakże główną wadą modeli eksponencjalnych jest ich zbyt duża złożoność obliczeniowa.

6.4.5. Modele adaptacyjne

W dotychczasowych rozważaniach język traktowany był jako obiekt homogeniczny. Niestety w praktyce język okazuje się być obiektem w wysokim stopniu heterogenicznym, którego podstawowe charakterystyki ulegają zmianie w przypadku zmiany tematyki dokumentu, jego stylu itp. W przypadku zastosowania aplikacji z dziedziny rozpoznawania języka naturalnego dla celów ogólnych, dane językowe napływają ze źródła, w oparciu o które model nie był wcześniej uczony. Zatem jedyną w pełni użyteczną informację o języku stanowi tylko i wyłącznie bieżący dokument. Najczęściej stosowaną technikę, pozwalającą na wyzyskanie informacji o języku zawartych w bieżącym dokumencie stanowi pamięć podręczna (ang. *cache*). W trakcie analizy dokumentu na bieżąco tworzony jest nowy model języka (zwykle w postaci n-gramu) oznaczany jako $P_{cache}(w|h)$. Natomiast aktualny model języka tworzony jest w następujący sposób

$$P_{adaptive}(w|h) = \lambda P_{static}(w|h) + (1-\lambda) P_{cache}(w|h) \tag{25}$$

6.5. PRZEWIDYWANE DALSZE KIERUNKI BADAŃ

Przyszłe kierunki badań w dziedzinie modelowania języków naturalnych będą się koncentrowały zapewne wokół zagadnień takich, jak np. redukcja liczby wymiarów przestrzeni poszukiwań. Jednym z przykładów interesującego obszaru badawczego jest słownictwo. Bowiem, w większości modeli języka słownictwo traktowane jest jedynie jako zbiór niepowiązanych ze sobą symboli. W modelach takich słowo BANK wcale nie jest bliżej słów POŻYCZKA lub BANKI, niż np. słowo BRAZYLIA. Z drugiej jednak strony intuicja podpowiada nam, że słownictwo języka stanowi złożoną strukturę. Zatem celem przyszłych prac musi być poznanie tej struktury i wbudowanie jej w przyszłe

modele języka. Nie jest to łatwe zadanie, a szczególne trudności upatrywane są w konstrukcji odpowiednich metryk, operujących na słowach i całych wyrażeniach, które pozwalałyby na wyrażenie w postaci liczbowej stopnia podobieństwa pomiędzy nimi.

7. PODSUMOWANIE

Artykuł niniejszy stanowi próbę wprowadzenia czytelnika w świat zagadnień translacji automatycznej. Translacja automatyczna posiada niemalże równie długą historię jak komputery. Była ona również jedną z pierwszych aplikacji, jakie zostały zaproponowane do zaimplementowania na maszynach cyfrowych. Jednakże dość szybko okazało się, że translacja maszynowa jest zadaniem o wiele trudniejszym, i zarazem o wiele ciekawszym, niż pierwotnie sądzono. W chwili obecnej można już powiedzieć, że zbudowanie systemu automatycznej translacji dla języka ogólnego, który pracowałby równie sprawnie jak człowiek, będący zawodowym tłumaczem nie jest możliwe – przynajmniej w oparciu o obecnie panującą technologię komputerowych technik przetwarzania danych. Translacja automatyczna wymaga bowiem inteligencji porównywalnej z inteligencją człowieka; wymaga również posiadania intuicji, ogólnej wiedzy o świecie oraz kierowania się zdrowym rozsądkiem, czyli cech umysłowości ludzkiej, których nie sposób sprowadzić do realizacji jakiegoś algorytmu. Nie oznacza to jednakże, że prowadzenie badań nad translacją automatyczną pozbawione jest sensu. Bowiem, dostępne już od wielu lat komercyjne systemy translacji automatycznej bardzo dobrze spełniają stawiane przed nimi zadania, pod warunkiem wszakże ograniczenia podawanych na ich wejście tekstów do pewnej wybranej dziedziny, pod kątem której system został zaprojektowany. Konieczne jest również nałożenie pewnych więzów na język, w którym formułowane są wypowiedzi podawane na wejście systemu, poprzez ograniczenie jego gramatyki oraz słownictwa. Takie postawienie sprawy pozwala na wyeliminowanie, przynajmniej w dużym stopniu, wieloznaczności, która jest cechą występującą powszechnie w każdym języku naturalnym i która jednocześnie sprawia najwięcej trudności podczas translacji automatycznej.

Na zakończenie artykułu wypada jeszcze raz podkreślić, że translacja maszynowa jest niezwykle pasjonującą, interdyscyplinarną dziedziną badań, która wymaga integracji wiedzy z różnych dziedzin, takich jak informatyka, językoznawstwo oraz filozofia języka, i która ciągle jeszcze wymaga przeprowadzenia licznych badań o charakterze podstawowym. Niestety, trzeba przyznać w tym miejscu z przykrością, że w Polsce translacja automatyczna jest dość mocno zaniedbanym kierunkiem badań, gdyż w wydawanych w Polsce czasopismach naukowych i technicznych oraz w materiałach konferencyjnych organizowanych w Polsce konferencji naukowo-technicznych, zgodnie z wiedzą autora, brak jest w ogóle jakichkolwiek artykułów na ten temat. Czas zatem ten stan rzeczy zmienić. Autor wyraża nadzieję, że niniejszy artykuł będzie krokiem postawionym we właściwym kierunku i pozwoli na wprowadzenie osób zainteresowanych w tą niezwykle ciekawą i pasjonującą dziedzinę wiedzy, jaką jest translacja automatyczna.

8. BIBLIOGRAFIA

1. J. R. Deller, J. G. Proakis, J. H. L. Hansen: *Discrete-time processing of speech signals*. New Jersey, Macmillan Publishing Company, USA, 1993.
2. J. Ratyńska: *Komputerowa analiza i synteza wypowiedzi użytkownika w systemie D&K*. Zakład Poligraficzny Politechniki Radomskiej, Monografie Nr 37, Radom, 1999.
3. B. Souvignier, A. Kellner, B. Rueber, H. Schramm: *The thoughtful elephant: Strategies for spoken dialog systems*. IEEE Transactions on Speech and Audio Processing, vol. 8, no. 1, January 2000, pp. 51–62.
4. V. Zue, S. Seneff, J. R. Glass, J. Polifroni, C. Pao, T. J. Hazen, L. Hetherington: *Jupiter: A telephone-based conversational interface for weather information*. IEEE Transactions on Speech and Audio Processing, vol. 8, no. 1, January 2000, pp. 85–95.
5. K. L. Baker, A. M. Franz, P. W. Jordan, T. Mitamura, E. H. Nyberg: *Coping with ambiguity in large-scale machine translation system*. Center for Machine Translation, Carnegie-Mellon University, Pittsburgh, PA, USA.
6. T. Mitamura, E. Nyberg, E. Torrejon, R. Igo: *Multiple strategies for automatic disambiguation in technical translation*. Center for Machine Translation, Carnegie-Mellon University, Pittsburgh, PA, USA.
7. L. Szczepaniak, Z. Królikowski: *Kontrolowane języki naturalne – przegląd rozwiązań i zastosowań*. Pro Dialog, nr 11, Poznań, Wydawnictwo Nakom, 2000, pp. 47–68.
8. C. Boitet: *Dialogue-based MT and self-explaining documents as an alternative to MAHT and MT of controlled languages*. Machine Translation Review, No. 10, October 1999, pp. 6–15.
9. J. Hutchins: *Computer-based translation systems and tools*. http://www.eamt.org/archive/hutchinsintro.html.
10. M. Vasconcellos: *Users of systems: The current state of MT usage or: How do I use thee? Let me count the ways*. Fourth Machine Translation Summit in Kobe, http://www.eamt.org/archive/summit93.html.
11. C. Brace: *Focus on Japan*. Language Industry Monitor, September/October 1993, http://www.eamt.org/archive/japan.html.
12. C. Brace, M. Vasconcellos, L. C. Miller: *MT users and usages: Europe and the Americas*. The fifth machine translation summit in Luxemburg, July 1995, http://www.eamt.org/archive/summit95.html.
13. Strona internetowa projektu Cordis, http://www.npk.gov.pl/cordis/www.cordis.lu/ist/projects/99-12256.htm.
14. N. Konĉar, D. Ŝipka: *Slavic as a source nad target language in NeuroTran sentence translation*. http://main.amu.edu.pl/sipkadan/fdsl2.htm.
15. Strona internetowa programu Poleng, http://poleng.wmid.amu.edu.pl/zasoby/tlumaczenie/english.html
16. P. Rutkowski: *Jak wiele można przetłumaczyć?* Wiedza i Życie, Luty 2001, ss. 42–44.
17. M. C. Corballis, J. L. Dessalles, R. Dunbar: *Aux origines du langage*. La Recherche, n^0 341, Avril 2001, pp. 27–39.
18. F. Majewicz: *Języki świata i ich klasyfikowanie*. Warszawa, PWN, 1989.
19. Melby: *Machine translation and philosophy of language*. Machine Translation Review, No. 9, April 1999, pp. 6–17.
20. D. Arnold, L. Balkan, S. Meijer, R. L. Humphreys, L. Sadler: *Machine translation: an introductory guide*. London, NCC Blachwell, 1994.
21. J. R. R. Leavitt, D. W. Lonsdale, K. Keck, E. H. Nyberg: *Tooling the lexicon acquisition process for large-scale KBMT*. Center for Machine Translation, Carnegie-Mellon University, Pittsburgh, PA, USA.
22. T. Mitamura: *Controlled language for multilingual machine translation*. Center for Machine Translation, Pittsburgh, Carnegie-Mellon University, PA, USA.
23. D. Z. Hakkani, G. Tür, T. Mitamura, E. H. Nyberg, K. Oflazer: *Issues in generating Turkish from interlingua*. Center for Machine Translation, Pittsburgh, Carnegie-Mellon University, PA, USA.

24. V. Cavalli-Sforza, K. Czuba, T. Mitamura, E. Nyberg: *Chalanges in adapting an interlingua for bidirectional English-Italian translatuion*. Center for Machine Translation, Pittsburgh, Carnegie-Mellon University, PA, USA.
25. J. G. Carbonell, T. Mitamura, E. H. Nyberg: *The KANT perspective: A critique of pure transfer (and pure interlingua, pure staistics,...)* Center for Machine Translation, Pittsburgh, Carnegie-Mellon University, PA, USA.
26. Waibel, P. Geutner, L. M. Tomokiyo, T. Schultz, M. Woszczyna: *Multilingualirty in speech and spoken language systems*. Proceedings of the IEEE, vol. 88, no. 8, August 2000, pp. 1297–1313.
27. W. Wiślicki: *Czy możliwy jest naukowy opis tego, co dzieje się w umyśle?* Wszechświat, t. 101, nr 1–3/2000, ss. 8–12.
28. R. Penrose: *Nowy umysł cesarza – o komputerach, matematyce i prawach fizyki*. Warszawa, PWN, 1995
29. S. Weinberg: *Czy uda się do roku 2050 zunifikować fizykę?* Świat Nauki, Styczeń 2000, ss. 38–45.
30. P. Kaeser: *Workflow automation of translation projects*. Proceedings of EAMT Workshop, Geneva, 1998, pp. 41–44.
31. M. S. Blekhman: *The PARS MT family: Practical usage*. Proceedings of EAMT Workshop, Geneva, 1998, pp. 45–53.
32. V. W. Zue, J. R. Glass: *Conversational interfaces: Advances and chalanges*. Proceedings of the IEEE, vol. 88, no. 8, August 2000, pp. 1166–1180.
33. Cegieła, A. Markowski: *Z polszczyzna za pan brat*. Warszawa, Wydawnictwo ISKRY, 1986.
34. Kreisberg, S. De Fanti: *Mówimy po włosku*. Warszawa, Wydawnictwo Wiedza Powszechna, 1992.
35. L. Stembor, S. Prędota: *Mówimy po niderlandzku*. Warszawa, Wydawnictwo Wiedza Powszechna, 1986.
36. J. Danecki: *Klasyczny język arabski*. Warszawa, Wydawnictwo Akademickie DIALOG, 1998.
37. M. Skwarczyński, M. Kordas, W. Zawadowski: *Leksykon matematyczny*. Warszawa, Wydawnictwo Wiedza Powszechna, 1995.
38. E. Bartz, A. Marlewski: *Komputerowe poszukiwanie rozwiązań pewnego równania diofantycznego*. Pro Dialog nr 10, Poznań, Wydawnictwo NAKOM, 2000, ss. 47–57.
39. R. Tadeusiewicz: *Możliwości i ograniczenia sieci neuronowych jako narzędzia w badaniach naukowych i w projektowaniu inżynierskim*. I Krajowa Konferencja: Metody i systemy komputerowe w badaniach naukowych i projektowaniu inżynierskim, Kraków, 1997, ss. II-1 – II-20.
40. R. Tadeusiewicz: *Problemy biocybernetyki*. Warszawa, PWN, 1994.
41. M. Blekhman, B. Pevzner: *First steps of language engineering in the USSR: The 50s through 70s*. Machine Translation Review, Issue No. 11, December 2000, pp. 5–7
42. D. Lonsdale: *Extraction d'un vocabulaire bilingue: outil et méthodes*. Center for Machine Translation, Pittsburgh, Carnegie-Mellon University, PA, USA.
43. J. Kitowski: *Współczesne systemy komputerowe*. Kraków, Wydawnictwo CCNS, 2000.
44. H. Ney, S. Niessen, F. J. Och, H. Sawaf, C. Tillmann, S. Vogel: *Algorithms for statistical translation of spoken language*. IEEE Transactions on Speech and Audio Processing, vol. 8, no. 1 , January 2000, pp. 24–35.
45. Waibel, P. Geatner, L. M. Tomokiyo, T. Schultz, M. Woszczyna: *Multilinguality in speech and spoken language systems*. Proceedings of the IEEE, vol. 88, no.8, August 2000, pp. 1297–1313.
46. H. Ney, S. Ortmanns: *Progress in dynamic programming search for LVCSR*. Proceedings of the IEEE, vol. 88, no. 8, August 2000, pp. 1224–1240.
47. L. K. Saul, M. G. Rahim: *Maximum likehood and minimum classification error factor analysis for automatic speech recognition*. IEEE Transactions on Speech and Audio Processing, vol. 8, no. 2, March, 2000, pp. 115–125.
48. R. Rosenfeld: *Two decades of statistical language modeling: where do we go from here?* Proceedings of the IEEE, vol. 88, no. 8, August 2000, pp. 1270–1278.

49. R. Damasio: *Jak rodzi się świadomość?* Świat Nauki, Styczeń 2000, ss. 76–81.
50. Strona internetowa systemu GEBMT, http://www.lti.cs.cmu.edu/Research/GEBMT
51. M. Nowy, R. Tadeusiewicz, J. Vetulani: *Rozmowy o mózgu.* Wszechświat, Tom 101, Nr 10–12, Październik-Listopad-Grudzień 2000, ss. 220–224.

M. GAJER

THE INTRODUCTION TO MACHINE TRANSLATION

Summary

Automatic translation between human languages called also machine translation MT is a long-term scientific dream of enormous social, political, and scientific importance. Machine translation was also one of the earliest applications suggested for digital computers, but turning this dream into reality has turned out to be a much harder, and in many ways a much interesting task than at first appeared. Now, machine translation is a reality. There are a lot of commercially available automatic translation systems, which very well can manage their tasks. In the paper the bases of machine translation systems are given and the history of their development is outlined. The special attention is paid to the issues that cause a lot of trouble during the automation of translation procepp. In the paper some modern machine translation systems are briefly characterised and the most promising research directions are mentioned. This author hopes that this paper will be a good introductory guide to the problems of machine translation, which will fill the gap in Polish scientific literature.

Keywords: natural language processing, computational linguistics, machine translation

Pracę wykonano w ramach grantu KBN: nr 8T11C 01917: „Formalne metody i narzędzia wspomagania analizy oraz projektowania baz danych i baz wiedzy"

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI

INFORMACJE DLA AUTORÓW

Redakcja przyjmuje do publikowania prace oryginalne, przeglądowe i monograficzne wchodzące w zakres szeroko pojętej elektroniki. Ponieważ KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI jest czasopismem Komitetu Elektroniki i Telekomunikacji Polskiej Akademii Nauk, w związku z tym na jego łamach znajdują się prace naukowe dotyczące podstaw teoretycznych i zastosowań z zakresu elektroniki, telekomunikacji, mikroelektroniki, optoelektroniki, radiotechniki i elektroniki medycznej.

Artykuły powinno charakteryzować oryginalne ujęcie zagadnienia, własna klasyfikacja, krytyczna ocena (teorii lub metod), omówienie aktualnego stanu, lub postępu danej gałęzi techniki oraz omówienie perspektyw rozwojowych.

Artykuły publikowane w innych czasopismach nie mogą być kierowane do druku w Kwartalniku Elektroniki i Telekomunikacji w drugiej kolejności zgłoszenia.

Objętość artykułu nie powinna przekraczać 30 stron po około 1800 znaków na stronie, w tym rysunki i tabele.

Wymagania podstawowe.

Artykuły należy nadsyłać na wyraźnym, jednostronnym, czarno-białym wydruku komputerowym. Wydruk w formacie A4 powinien mieć znormalizowaną liczbę wierszy i znaków w wierszu (30 wierszy po 60 znaków w wierszu), w dwóch egzemplarzach, w języku polskim lub angielskim wybranym przez autora. Do wydruku powinna być dołączona dyskietka z elektronicznym tekstem artykułu. Preferowane edytory to WORD 6 lub 8.

Układ artykułu (w wersji podstawowej) musi być następujący:

- Tytuł.
- Autor (imię i nazwisko autora/ów).
- Miejsce pracy (nazwa instytucji, miejscowość, adres, + ew. adres elektroniczny (e-mail)).
- Zwięzłe streszczenie powinno być w języku takim, w jakim jest pisany artykuł (wraz ze słowami kluczowymi).
- Tekst podstawowy powinien mieć następujący układ:

1. WPROWADZENIE
2. np. TEORIA
3. np. WYNIKI NUMERYCZNE

3.1. .....

3.2. .....

4. .....
5. .....
6. PODSUMOWANIE
7. ew. PODZIĘKOWANIE
8. BIBLIOGRAFIA

- Układ streszczenia w dodatkowej wersji językowej powinien być następujący;

TYTUŁ (w języku angielskim – o ile artykuł pisany jest w języku polskim i na odwrót),
AUTOR (inicjał imienia i nazwisko).

Obszerne do 3600 znaków streszczenie (wraz z słowami kluczowymi) w języku:

a) angielskim, gdy artykuł pisany jest w języku polskim,
b) polskim, gdy artykuł pisany jest w języku angielskim.

Streszczenie to powinno pozwolić czytelnikowi na uzyskanie istotnych informacji zawartych w pracy.

Z tego względu w streszczeniu tym mogą być cytowane numery istotnych wzorów, rysunków i tabel zawartych w podstawowej wersji językowej.

- Wszystkie strony muszą mieć numerację ciągłą.

Sposób pisania tekstu.

Tekst powinien być pisany bez używania wyróżnień, a w szczególności nie dopuszcza się spacjowania, podkreślania i pisania tekstu dużymi literami z wyjątkiem wyrazów, które umownie pisze się dużymi literami

(np. FORTRAN). Proponowane wyróżnienia Autor może zaznaczyć w maszynopisie zwykłym ołówkiem za pomocą przyjętych znaków adiustacyjnych np. podkreślenie linią przerywaną oznacza spacjowanie (rozstrzelenie), podkreślenie linią ciągłą – pogrubienie, podkreślenie wężykiem – kursywa.
Tekst powinien być napisany z podwójnym odstępem między wierszami, tytuły i podtytuły małymi literami. Marginesy z każdej strony powinny mieć około 35 mm. Wielkość czcionki wydruku powinna być zbliżona co najmniej do wielkości czcionki maszyny do pisania (minimum 12 punktów). Przy podziale pracy na rozdziały i podrozdziały cyfrowe ich oznaczenia nie powinny być większe niż III stopnia (np. 4.1.1.).

Sposób pisania tabel.
Tabele powinny być pisane na oddzielnych stronach. Tytuły rubryk pionowych i poziomych powinny być napisane małymi literami z podwójnym odstępem między wierszami. Przypisy (notki) dotyczące tabel należy pisać bezpośrednio pod tabelami. Tabele należy numerować kolejno liczbami arabskimi, u góry każdej tabeli podać tytuł dwujęzyczny. W pierwszej kolejności w podstawowej wersji językowe, a później w dodatkowej wersji językowej. Tabele umieścić na końcu maszynopisu. Przyjmowane są tabele algorytmów i programy na wydrukach komputerowych. W tym przypadku zachowany jest ich oryginalny układ. Tabele powinny być cytowane w tekście.

Sposób pisania wzorów matematycznych.
Rozmieszczenie znaków, cyfr, liter i odstępów powinno być zbliżone do rozmieszczenia elementów druku. Wskaźniki i wykładniki potęg powinny być napisane wyraźnie i być prawidłowo obniżone lub podwyższone w stosunku do linii wiersza podstawowego. Znaki nad literami i cyframi, całkami i in. symbolami (strzałki, linie, kropki, daszki) powinny być umieszczone dokładnie nad tymi elementami, do których się odnoszą. Numery wzorów cyframi arabskimi powinny być kolejne i umieszczone w nawiasach okrągłych z prawej strony. Nazwy jednostek, symbole literowe i graficzne powinny być zgodne w wytycznymi IEE (Intemational Electronical Commision) oraz ISO (Intemational Organization of Standarization).

Powołania.
Powołania na publikacje powinny być umieszczone na ostatnich stronach tekstu pod tytułem „Bibliografia", opatrzone numeracją kolejną bez nawiasów. Numeracja ta powinna być zgodna z odnośnikami w tekście artykułu. Przykłady opisu publikacji:

- periodycznej
- nieperiodyczne
- książki

1. F. Valdoni: A new milimetre wave satelitę. E.T.T. 1990, vol. 2, no 5, pp. 141–148
2. K. Andersen: A resource allocation framework. XVI Intemational Symposium, Stockholm (Sweden), may 1991, paper A 2.4
3. Y.P. Tvidis: Operation and modeling ofthe MOS transistors. New York, McGraw-Hill, 1987, p. 553

Materiały ilustracyjne.
Rysunki powinny być wykonane wyraźnie, na papierze gładkim lub milimetrowym w formacie nie mniejszym niż 9×12 cm. Mogą być także w postaci wydruku komputerowego (preferowany edytor Corel Draw). Fotografie lub diapozytywy przyjmowane są raczej czarno-białe w formacie nie przekraczającym 10×15 cm. Na marginesie każdego rysunku i na odwrocie fotografii powinno być napisane ołówkiem imię i nazwisko Autora oraz skrót tytułu artykułu, do którego są przeznaczone oraz numer rysunku. Spis podpisów pod rysunki i fotografie powinien być umieszczony na oddzielnej stronie. Podpisy pod rysunkami (fotografiami) powinny być dwujęzyczne: w pierwszej kolejności w podstawowej wersji językowej, a później w dodatkowej wersji językowej. Rysunki powinny być cytowane w tekście.

Uwagi dodatkowe.
Na odrębnej stronie powinny być podane następujące informacje:
- adres do korespondencji z kodem pocztowym (domowy lub do miejsca pracy),
- telefon domowy i/lub do miejsca pracy,
- adres e-mailowy (jeśli autor posiada).

Autorowi przysługuje bezpłatnie 20 odbitek artykułu. Dodatkowe egzemplarze odbitek, lub cały zeszyt Autor może zamówić u wydawcy na własny koszt.

Autora obowiązuje korekta autorska, którą powinien wykonać w ciągu 3 dni od daty otrzymania tekstu z Redakcji oraz zwrócić osobiście, lub listownie pod adres Redakcji. Korekta powinna być naniesiona na przekazanych Autorowi szpaltach na marginesach ew. na osobnym arkuszu w przypadku uzupełnień tekstu większych niż dwa wiersze. W przypadku nie zwrócenia korekty w terminie, korektę przeprowadza Redakcja Techniczna Wydawcy.

Redakcja prosi Autorów o powiadomienie ją o zmianie miejsca pracy i adresu prywatnego.

INFORMATION FOR AUTHORS OF K.E.T

The editorial stuff will accept for publishing only original monographic and survey papers concerning widely understood electronics. Because of the fact that KWATRALNIK ELEKTRONIKI I TELEKOMUNIKACJI is a journal of the Committee for Electronics and Telecommunications of Polish Academy of Science, it presents scientific works concerning theoretical bases and applications from the field of electronics, telecommunications, microelectronics, optoelectronics, radioelectronics and medical electronics.
Articles should be characterised by original depiction of a problem, its own classification, critical opinion (concerning theories or methods), discussion of an actual state or a progress of a given branch of a technique and discussion of development perspectives.
An article published in other magazines can not be submitted for publishing in K.E.T.
The size of an article can not exceed 30 pages, 1800 character each, including figures and tables.

Basic requirements
The article should be submitted to the editorial stuff as a one side, clear, black and white computer printout in two copies. The article should be prepared in English or Polish. Floppy disc with an electronic version of the article should be enclosed. Preferred wordprocessors: WORD 6 or 8.
Layout of the article.
- Title.
- Author (first name and surname of author/authors).
- Workplace (institution, address and e-mail).
- Concise summary in a language article is prepared in (with keywords).
- Main text with following layout:
  - Introduction
  - Theory (if applicable)
  - Numerical results (if applicable)
  - Paragraph 1
  - Paragraph 2
  - .....
  - .....
  - Conclusions
  - Acknowledgements (if applicable),
  - References
- Summary in additional language:
  - Title (in Polish, if article was prepared in English and vice versa)
  - Author (firs name initials and surname)
  - Extensive summary, hawever not exceeching 3600 characters (along with keywords) in Polish, if artide wos prepared in English and vice versa). The summary should be prepared in a way allowing a reader to obtaoin essential information contained in the artide. For that reason in the summary author can place numbers of essential formulas, figures and tables from the article.

Pages should have continues numbering.

Main text
Main text can not contain formatting such as spacing, underlining, words written in capital letters (except words that are commonly written in capital letters). Author can mark suggested formatting with pencil on the margin of the article using commonly accepted adjusting marks.
Text should be written with double line spacing with 35 mro left and right margin. Titles and subtitles should be written with small letters. Titles and subtitles should be numbered using no more than 3 levels (i.e. 4.1.1.).

Tables
Tables with their titles should be placed on separate page at the end of the article. Titles of rows and columns

should be written in small letters with double line spacing. Annotations concerning tables should be placed directly below the table. Tables should be numbered with Arabic numbers on the top of each table. Table can consist algorithm and program listings. In such cases original layout of the table will be preserved. Table should be cited in the text.

Mathematical formulas

Characters, numbers, letters and spacing of the formula should be adequate to layout of main text. Indexes should be properly lowered or raised above the basic line and clearly written. Special characters such as lines, arrows, dots should be placed exactly over symbols which they are attributed to. Formulas should be numbered with Arabic numbers place in brackets on the right side of the page. Units of measure, letter and graphic symbols should be printed according to requirements of IEE (International Electronical Commission) and ISO (International Organisation of Standardisation).

References

References should be placed at the and of the main text with the subtitle „References". References should be numbered (without brackets) adequately to references placed in the text. Examples of periodical [1], non-periodical [2] and book [3] references:

1. F. Valdoni: A new millimetre wave satelite. E.T.T. 1990, vol.2, no 5, pp. 141–148
2. K. Anderson: A recource allocation framework. XVI International Symposium, Stockholm (Sweden), May 1991. paper A 2.4
3. Y.P. Tvidis: Operation and modelling of the MOS transistors. New York, McGraw-Hill, 1987, p. 553

Figures

Figures should be clearly drawn on plain or millimetre paper in the format not smaller than 9×12 cm. Figures can be also printed (preferred editor – CorelDRAW). Photos or diapositives will be accepted in black and white format not grater than 10×15 cm. On the margin of each drawing and on the back side of each photo author name and abbreviation of the title of article should be placed. Figure's captures should be given in two longuaqes (first in the languaqe the article is writtes in and then in additional larguaqe). Figure's captures shoukl be also listed on separate page. Figures should be cited inthe text.

Additional information

On the separate page following information should be placed:
- mailing address (home or office),
- phone (home or/and office),
- e-mail.

Author is entitled to free of charge 20 copies of article. Additional copies or the whole magazine can be ordered at publisher at the one's expense.

Author is obliged to perform the author's correction, which should be accomplished within 3 days starting from the date of receiving of the text from the editorial stuff. Corrected text should be return to the editorial stuff personally or by mail. Correction marks should be placed on the margin of copies received from the editorial stuff or if needed on separate pages. In the case when the correction is not returned in said time limit, correction will be performed by technical editorial stuff of the publisher.

In case of changing of workplace or home address Authors are asked to inform the editorial stuff.

Subscritpion for external subscribers:

The promotional subscription price in 2002 is $ 100 including postage for institutions. At subscriber's request this journal will be air maild at additional postage to 50% dross proce to European countries nad 65% overseas.

– Foreign Trade Enterprise ARS Polona, Krakowskie Przedmieście 7, 00-068 Warszawa, Poland,

– RUCH S.A. tel. (4822) 53-28-819, 53-28-823, fax 53-28-731.

Pragnę Państwa zawiadomić, że można już zamawiać prenumeratę czasopism naukowych w Wydawnictwie Naukowym PWN. Z pewnością usprawni to obsługę prenumeratorów i pozwoli zebrać informacje o czytelnikach.

Zamówienia można składać telefonicznie pod numerami (022) 695-42-74, (022) 695-40-47, przysłać faksem pod nr (022) 695-42-70, lub listownie (wzór zamówienia jest dostępny w Dziale Publikacji Zleconych Wydawnictwa Naukowego S.A.).