ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1998
Seria: AUTOMATYKA z.122 Nr kol. 1388

Zygmunt KUŚ

PROJEKTOWANIE FILTRÓW STOSOWYCH NA PODSTAWIE KRYTERIUM MINIMALIZACJI WARTOŚCI BŁĘDU BEZWZGLĘDNEGO

Streszczenie. W pracy została przedstawiona klasa filtrów stosowanych posiadających właściwość układania w stos i wykorzystujących progową dekompozycję. Klasa ta zawiera filtry oparte na statystykach porządkowych (w szczególności filtr medianowy), filtry morfologiczne i wiele innych.

Jako kryterium wyboru filtru stosowego przyjmuje się średnią wartość bezwzględną pomiędzy sygnałem idealnym a wyjściem filtru. Minimalizacja średniego błędu bezwzględnego może być uzyskana jako rozwiązanie odpowiedniego zadania programowania liniowego.

Przedstawione, rozwinięte metody pozwalają na obliczanie współczynników macierzy ograniczeń dla wcześniej sformułowanego zadania programowania liniowego.

Na koniec przedstawiono przykłady optymalnych filtrów stosowych.

STACK FILTER DESIGN BASED ON THE MEAN ABSOLUTE ERROR CRITERION

Summary. The class of stack filters preserving stacking property and threshold decomposition is introduced. Rank order filters (especially median filters), morphological filters and many others are included in this class.

As a criteria for selection of a stack filter the minimization of the mean absolute error between required signal and the filter output is taken. The minimization of mean absolute error can be obtained through solution of the appropriate linear programming problem.

These enhanced methods allow for computation of the coefficients and the matrix of constraints for the linear programming problem formulated above.

Finally, examples of stack filters are presented.

Wstęp

W dziedzinie przetwarzania obrazów występują obecnie trendy zmierzające z jednej strony do poszukiwania nowych algorytmów przetwarzania, a z drugiej do poszukiwania nowych sposobów realizacji znanych już algorytmów. Powodem tych poszukiwań jest dążenie do przyspieszenia działania układów i oprogramowania oraz zmniejszenia kosztów produkcji.

Szczególnie duże wymagania, jeśli chodzi o szybkość działania, stawia przetwarzanie obrazów w czasie rzeczywistym.

Poszukiwania możliwości zwiększenia szybkości działania oraz uproszczenia konstrukcji powodują, że podejście do przetwarzania obrazów oferowane przez metodę przetwarzania stosowego jest szczególnie atrakcyjne.

Klasę filtrów stosowych definuje się jako posiadającą właściwość słabej superpozycji, nazywaną progową dekompozycją i właściwość porządkującą, nazywaną właściwością układania w stos. W klasie filtrów stosowych zawierają się filtry ze statystykami porządkowymi oraz filtry składające się ze skończonej liczby filtrów ze statystykami porządkowymi, wszystkie filtry morfologiczne oparte na operacjach „otwierania" i „zamykania" oraz wiele innych szeroko stosowanych filtrów.

Metoda pzetwarzania stosowego polega na dekompozycji ciągu wielowartościowego na ciągi binarne, wykonaniu operacji przetwarzania na tych ciągach, co jest szczególnie proste, a następnie powrocie do ciągu wielowartościowego przez „sumowanie" wynikowych ciągów binarnych.

Różne rodzaje filtracji uzyskujemy przez zastosowanie na każdym poziomie progowania pewnej funkcji boolowskiej. O filtracji stosowej mówimy w momencie, gdy zastosowana funkcja boolowska posiada właściwość układania w stos. Funkcja boolowska posiada właściwość układania w stos, jeżeli w swojej zminimalizowanej postaci sumacyjnej nie zawiera dopełnień zmiennych wejściowych. Funkcje takie nazywamy dodatnimi funkcjami boolowskimi.

Wspomniana dekompozycja pozwala na realizację znanych algorytmów przetwarzania w odmienny niż do tej pory sposób. Istotną cechą tego podejścia jest łatwość wykonania przetworników stosowych przy wykorzystaniu technologii VLSI, co umożliwia wzrost szybkości działania, pozwalający zrealizowanie przetwarzania obrazów w czasie rzeczywistym.

Kolejną ważną cechą przetworników stosowych jest możliwość konstrukcji dużej liczby różnych przetworników. Właściwości wielu z nich nie zostały dotychczas jeszcze zbadane. Dla ilustracji tej różnorodności powiedzmy tylko, że przetworników z aperturą szerokości 3 jest 20, z aperturą 5 jest 7581, a z aperturą 7 jest co najmniej 2^{35}.

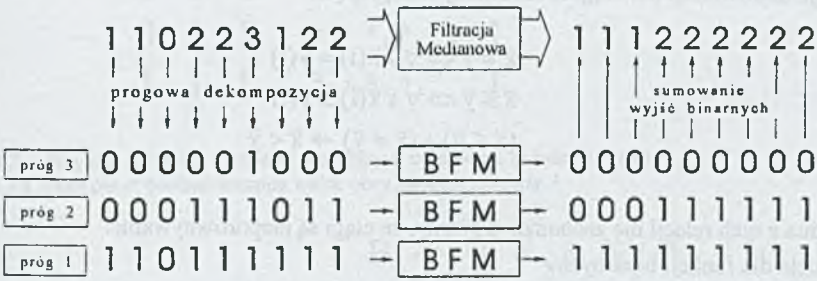
Przy tak licznej klasie przetworników stosowych pojawia się problem dopasowania rodzaju przetwarzania do konkretnej sytuacji, czyli problem wyboru funkcji boolowskiej określającej dany filtr.

Zatem pierwszym zadaniem, jakie trzeba sobie postawić, jest określenie kryterium, według którego będziemy poszukiwać funkcji boolowskiej definiującej filtr stosowy. W niniejszej pracy rozważa się sytuację, gdy minimalizowanym wskaźnikiem jest wartość średnia błędu bezwzględnego.

Z omawianiem podstawowych właściwości przetwarzania stosowego można się spotkać w [1,3,4,5,6]. Problemami związanymi z przeszukiwaniem funkcji boolowskiej minimalizującej średni błąd bezwzględny zajmowali się między innymi Coyle Edward i Lin Jean-Hsang w [2].

1. Podstawowe definicje i właściwości przetwarzania stosowego

Przedstawione zostaną tutaj podstawowe definicje i twierdzenia dotyczące przetwarzania stosowego. Idea przetwarzania stosowego została zilustrowana na przykładzie realizacji przetwarzania medianowego - rys.1.1.



$$BFM = X_1X_2 + X_1X_3 + X_2X_3$$

Rys.1.1. Filtracja medianowa wykonana metodą przetwarzania stosowego
Fig.1.1. Median filtering by stack filtering

Na rys.1.1 widzimy, że przetwarzanie stosowe polega na wkonaniu progowej dekompozycji czterowartościowego ciągu wejściowego do trzech ciągów binarnych, binarnej filtracji medianowej tych ciągów binarnych i sumowaniu ciągów binarnych, co daje w efekcie czterowartościowy ciąg wyjściowy.

Progową dekompozycję definiujemy w następujący sposób:

Definicja progowej dekompozycji

Progową dekompozycją M - wartościowego ciągu R o elementach R(k) nazywamy zbiór M-1 binarnych ciągów, zwanych ciągami sprogowanymi:

$$T_1, T_2, ..., T_{M-1}, \tag{1}$$

których elementy określamy następująco:

$$T_i(j) = \begin{cases} 1 \text{ jeśli } R(j) \geq i \\ 0 \text{ jeśli } R(j) < i \end{cases} \tag{2}$$

Sumowanie wyjściowych ciągów binarnych dla większej liczby poziomów wymagałoby przy realizacji w technologii VLSI wykonania bardzo dużego sumatora, dlatego omija się te konieczność dzięki założeniu, że funkcja boolowska posiada właściwość układania w stos, definiowaną następująco:

Definicja właściwości układania w stos

Definicja dla uporządkowanych zbiorów ciągów:

Uporządkowany zbiór M binarnych ciągów $S_1, S_2, ... S_M$ poiada właściwość układania w stos, jeżeli:

$$S_1 \geq S_2 \geq .....\geq S_M \tag{3}$$

Relacje nierówności dla ciągów określamy następująco:

$$\begin{aligned} &\bar{x} = \bar{y} \Leftrightarrow \forall\, i\ \bar{x}(i) = \bar{y}(i) \\ &\bar{x} \leq \bar{y} \Leftrightarrow \forall\, i\ \bar{x}(i) \leq \bar{y}(i) \\ &(\bar{x} \leq \bar{y}) \wedge (\bar{x} \neq \bar{y}) \rightarrow \bar{x} < \bar{y} \end{aligned} \tag{4}$$

Jeżeli żadna z tych relacji nie zachodzi, mówimy, że ciągi są nieporównywalne.

Definicja dla funkcji binarnych:

O funkcji binarnej B(·) mówimy, że posiada właściwość układania w stos, jeżeli:

$$\bar{x} \geq \bar{y} \Rightarrow B(\bar{x}) \geq B(\bar{y}) \tag{5}$$

Rozważając zapis funkcji boolowskiej w postaci wyrażenia możemy sformułować warunek posiadania przez funkcję właściwości układania w stos.

Warunek konieczny i wystarczający posiadania przez funkcję binarną właściwości układania w stos

Binarna funkcja posiadania właściwości układania w stos ma miejsce wtedy i tylko wtedy, jeżeli może być wyrażona w postaci wyrażenia boolowskiego nie zawierającego żadnych dopełnień zmiennych wejściowych [1].

Na rys.1.2.a.b przedstawiono przykładowe funkcje posiadające i nie posiadające właściwość układania w stos. Wektory zawierające argumenty funkcji boolowskiej [$\bar{x}$(1), $\bar{x}$(2), $\bar{x}$(3)] uszeregowano od największego do najmniejszego (4), od dołu do góry. Zatem odpowiadajace im wyjscia funkcji powinny dać kolumnę mającą od dołu ciąg jedynek i idąc dalej

w górę następujący za nim ciąg zer i tak się dzieje w przykładzie z rys.1.2.a, gdzie mamy do czynienia z dodatnią funkcją boolowską.

Jeżeli zauważymy, ze operacja progowania daje nam zawsze w efekcie uporządkowany zbiór ciągów posiadający właściwość układania w stos oraz założymy użycie w miejsce BFM z rys.1.1 funkcji boolowskiej również posiadającej tę właściwość, to widzimy, że wyjściowe ciągi binarne również tworzą uporządkowany zbiór, posiadający właściwość układania w stos. Dla danej chwili czasu (numeru elementu) w ciągu wejściowym otrzymujemy, idąc od największego poziomu progowania, kolejno same zera i same jedynki. Stąd widzimy, że sumowanie można zastąpić przeprowadzeniem detekcji poziomu, na którym nastąpiła pierwsza zmiana z 0 na 1.

$$B(\cdot) = x_2 + x_1 \cdot x_3$$

x_1	x_2	x_3		
0	0	0	$x_2 + x_1 \cdot x_3 =$	0
0	0	1	$x_2 + x_1 \cdot x_3 =$	0
0	1	1	$x_2 + x_1 \cdot x_3 =$	1
1	1	1	$x_2 + x_1 \cdot x_3 =$	1

Rys.1.2.a. Przykład funkcji boolowskiej posiadającej właściwość układania w stos
Fig.1.2.a. Example of boolean function which obey stacking property

$$B(\cdot) = \bar{x}_2 + x_1 \cdot x_3$$

x_1	x_2	x_3		
0	0	0	$\bar{x}_2 + x_1 \cdot x_3 =$	1
0	0	1	$\bar{x}_2 + x_1 \cdot x_3 =$	1
0	1	1	$\bar{x}_2 + x_1 \cdot x_3 =$	0
1	1	1	$\bar{x}_2 + x_1 \cdot x_3 =$	1

Rys.1.2.b. Przykład funkcji boolowskiej nie posiadającej właściwości układania w stos
Fig.1.2.b. Example of boolean function which not obey stacking property

Funkcje boolowskie posiadające właściwość układania w stos nazywane są dodatnimi funkcjami boolowskimi. Nieznana jest dokładna analityczna zależność pomiędzy liczbą dodatnich funkcji boolowskich a szerokoscią apertury filtru, wiadomo natomiast, że dla 3 punk-

tów w aperturze jest tych funkcji 20, dla 5 jest ich 7581, a dla apertury szerokości 7 jest co najmniej 2^{35} dodatnich funkcji boolowskich.

Widzimy zatem, że wybranie jednej z wielu możliwych funkcji wymaga zastanowienia się, jakim kryterium można się w tym wyborze kierować. W rozdziale 2 będzie sformułowane kryterium wartości średniej błędu bezwzględnego i będziemy poszukiwać funkcji boolowskiej minimalizującej ten wskaźnik ale najpierw zostanie przedstawiona statystyczna interpretacja funkcji boolowskiej i właściwości układania w stos.

Definicja stochastycznej funkcji boolowskiej

Zakładamy, że 2^n możliwych ciągów binarnych o długości n mamy uporządkowanych i oznaczonych jako $\overline{x}_1, \overline{x}_2, ..., \overline{x}_{2n}$.

Stochastyczną funkcję boolowską n - zmiennych B(·) definujemy wektorem $\overline{P}_B$ z 2^n elementami, w którym i - ty element oznacza:

$$P_B(1 \mid \overline{x}_i) = P\,(\text{B ma wartość } 1 \mid \overline{x}_i \text{ jest argumentem B}) \tag{6}$$

Jednocześnie

$$P_B(0 \mid \overline{x}_i) = 1 - P_B(1 \mid \overline{x}_i) \tag{7}$$

Definicja stochastycznej właściwości układania w stos

Stochastyczna funkcja boolowska posiada stochastyczną właściwość układania w stos, jeżeli:

$$E(B(\overline{x})) \geq E(B(\overline{y})) \Leftrightarrow \overline{x} \geq \overline{y}, \tag{8}$$

gdzie E jest operatorem wartości oczekiwanej.

Zadanie poszukiwania funkcji boolowskiej minimalizującej średni błąd bezwzględny zostanie rozwiązane przez podanie metody znajdowania stochastycznej funkcji boolowskiej minimalizującej średni błąd bezwzględny, a następnie wykazane zostanie, że prawdopodobieństwa definiujące tę funkcję przyjmują zawsze wartości 0 lub 1. Zatem uzyskiwana stochastyczna funkcja boolowska równoważna jest deterministycznej funkcji boolowskiej, którą wykorzystujemy do stworzenia filtru stosowego.

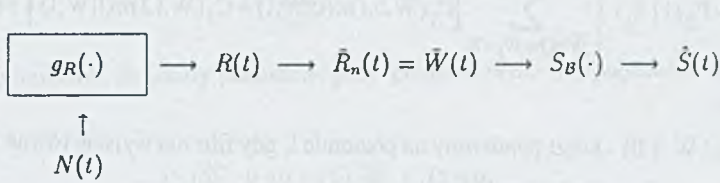
2. Sformułowanie zadania filtracji optymalnej

W rozdziale tym zostanie przedstawiony model zadania optymalizacji doboru filtru z wyszczególnieniem uwzględnianych sygnałów, określony minimalizowany wskaźnik jakości

oraz sformułowane zadanie minimalizacji średniego błędu bezwzględnego jako zadanie programowania liniowego.

2.1. Model zadania optymalizacji

Na rys.2.1 pokazany jest model zadania filtracji optymalnej. Sygnał oryginalny (niezakłócony) S (t) jest zaszumiany sygnałem n (t). Sposób tego zaszumiania opisuje statyczna funkcja $g_R(\cdot)$; wartość R(t) w chwili t zależy tylko od wartości S(t) i N(t) w chwili t. Z sygnału R(t) wybierany jest ciąg $\overline{R}_n$ (t) o długości n, równej szerokości apertury filtru $S_B(\cdot)$. Ciąg ten nazywany będzie wektorem aperturowym i oznaczany jako $\overline{W}$(t). W wyniku przetwarzania przez filtr stosowany S_B (·) sygnału wektora aperturowego uzyskujemy ocenę sygnału oryginalnego $\hat{S}$(t).



$S(t)$	-	sygnał idealny	$\bar{W}(t)$	-	wektor aperturowy
$N(t)$	-	sygnał zakłócający	$\hat{S}(t)$	-	sygnał wyjściowy z filtru
$R(t)$	-	sygnał obserwowany przez filtr	$S_B(\cdot)$	-	filtr stosowy
n	-	szerokość apertury filtru			

Rys.2.1. Model zadania filtracji optymalnej
Fig.2.1. Optimal filtration problem model

Szczegółowy opis modeli sygnałów znajduje się w rozdziale 3.1.

2.2. Wskaźnik jakości

Poszukujemy funkcji boolowskiej minimalizującej średni błąd bezwzględny.

$$C(S_B) = E[\,|\,S(t) - S_B(\overline{W}(t))\,|\,] \tag{9}$$

Dzięki właściwości układania w stos otrzymujemy:

$$C(S_B) = \sum_{i=1}^{M} E[\,|\,s_i(t) - B(\overline{w}_i(t))\,|\,], \tag{10}$$

gdzie M jest liczbą poziomów progowania, $s_i(t)$ sygnałem idealnym sprogowanym na poziomie i, a $\overline{w}_i(t)$ jest wektorem aperturowym $\overline{W}(t)$ sprogowanym na poziomie i.

Dzięki sformułowaniu oceny sygnału oryginalnego przez filtr na podstawie sygnału zaszumionego jako problemu decyzyjnego możemy przedstawić wskaźnik kosztów ponoszonych na poziomie progowania l przy użyciu funkcji boolowskiej B w następującej postaci:

$$C(B \mid l) =$$

$$\sum_{j=1}^{2^n} \left[P_B(0 \mid \overline{x}_j) \cdot \left(\sum_{\overline{W} \in Qw : \overline{w}_l = \overline{x}_j} \left[C_l(\overline{W},0,0)\pi(0|\overline{W},l) + C_l(\overline{W},1,0)\pi(1|\overline{W},l)\right] \cdot \pi(\overline{W}) \right) + \right.$$

$$\left. + P_B(1 \mid \overline{x}_j) \cdot \left(\sum_{\overline{W} \in Qw : \overline{w}_l = \overline{x}_j} \left[C_l(\overline{W},0,1)\pi(0|\overline{W},l) + C_l(\overline{W},1,1)\pi(1|\overline{W},l)\right] \cdot \pi(\overline{W}) \right) \right],$$

gdzie:

$C_l(\overline{W},1,0)$ - koszt ponoszony na poziomie l, gdy filtr ma wyjście równe 0 a w oryginalnym sygnale jest 1,

$C_l(\overline{W},0,0)$ - wyjściem filtru jest 0, powinno być 0,

$C_l(\overline{W},0,1)$ - wyjściem filtru jest 1, powinno być 0,

$C_l(\overline{W},1,1)$ - wyjściem filtru jest 1, powinno być 1,

$\pi(0 \mid \overline{W}, l) = 1 - \pi(1 \mid \overline{W}, l)$ = prawd. (poprawne wyj. filtru = 1 | na wejściu jest $\overline{W}$).

Całkowity koszt ponoszony przy użyciu w filtrze stosowym funkcji boolowskiej B otrzymujemy w wyniku sumowania kosztów ponoszonych na poszczególnych poziomach progowania.

$$C(B) = \sum_{i=1}^{M} C(B \mid l) \tag{11}$$

Przyjmując

$$C^1 = \left(\sum_{\overline{W} \in Q_w : \overline{w}_l = \overline{x}_j} \left[C_l(\overline{W},0,0)\pi(0|\overline{W},l) + C_l(\overline{W},1,0)\pi(1|\overline{W},l)\right] \cdot \pi(\overline{W}) \right) \tag{12}$$

$$C^2 = \left(\sum_{\overline{W} \in Q_w : \overline{w}_l = \overline{x}_j} \left[C_l(\overline{W},0,1)\pi(0|\overline{W},l) + C_l(\overline{W},1,1)\pi(1|\overline{W},l)\right] \cdot \pi(\overline{W}) \right) \tag{13}$$

oraz korzystając z zależności

$$P_B(0 \mid \overline{x}_i) = 1 - P_B(1 \mid \overline{x}_i) \tag{14}$$

otrzymujemy:

$$C(B \mid 1) = \sum_{j=1}^{2^n} P_B(1 \mid \overline{x}_i) \cdot (C^2 - C^1) + C^1 \tag{15}$$

Ponieważ wpływ wyrazu wolnego C^1 na wartość wskaźnika C (B | 1) nie zależy od P_B (1 | $\overline{x}_i$) definiujących poszukiwaną funkcję boolowską, możemy zmodyfikować nasz wskaźnik do postaci:

$$C'(B \mid 1) = \sum_{j=1}^{2^n} P_B(1 \mid \overline{x}_i) \cdot (C^2 - C^1) \tag{15}$$

Następnie przyjmujemy, że koszty ponoszone przy podjęciu przez filtr poprawnej decyzji są równe 0

$$C_1(\overline{W}, 0, 0) = C_1(\overline{W}, 1, 1) = 0 \tag{17}$$

i podstawiając

$$\pi(1 \mid \overline{W}, 1) = 1 - \pi(0 \mid \overline{W}, 1) \tag{18}$$

otrzymujemy:

$$C'(B \mid 1) = \sum_{j=1}^{2^n} P_B(1 \mid \overline{x}_j) \cdot \left[\sum_{\overline{W} \in Q_w : \overline{w}_1 = \overline{x}_j} \pi(\overline{W}) \cdot \left\{ \pi(0 \mid \overline{W}, 1) \cdot \left[C_1(\overline{W}, 0, 1) + C_1(\overline{W}, 1, 0) \right] - C_1(\overline{W}, 1, 0) \right\} \right] \tag{19}$$

$$C'(B) = \sum_{i=1}^{M} C'(B \mid 1) \tag{20}$$

Przyjmując: koszty C_1 ($\overline{W}$,0,1) = CW01 i C_1 ($\overline{W}$,1,0) = CW10 niezależne od poziomu progowania i wektora aperturowego oraz binarny sygnał R (t) na wejściu (1 = 1) otrzymujemy:

$$C'(B) = \sum_{j=1}^{2^n} P_B(1 \mid \overline{x}_j) \cdot \left[\pi(\overline{x}_j) \cdot \left\{ \pi(0 \mid \overline{x}_j) \cdot \left[CW01 + CW10 \right] - CW10 \right\} \right], \tag{21}$$

gdzie

$$\pi(\overline{x}_j) = \sum_{\overline{W} \in Q_w : \overline{w}_l = \overline{x}_j} \pi(\overline{W}) \tag{22}$$

$$\pi(0 \mid \overline{x}_j) = \sum_{\overline{W} \in Q_w : \overline{w}_l = \overline{x}_j} \pi(0 \mid \overline{W}) \tag{23}$$

2.3. Sformułowanie zadania minimalizacji średniego błędu bezwzględnego jako zadania programowania liniowego

Zauważmy, że uzyskany w poprzednich rozważaniach wskaźnik C' (B) jest liniową kombinacją prawdopodobieństw określających poszukiwaną funkcję boolowską i pewnych współczynników, stałych dla przyjętego modelu sygnałów S (t) i N (t) i dla ustalonych kosztów CW01 i CW10.

Oznaczając wektor prawdopodobieństw definiujących funkcję boolowską jako

$$P = [P_1, P_2, ..., P_{2n}], \tag{24}$$

gdzie

$$P_j = P_B(1 \mid \overline{x}_j) \tag{25}$$

oraz wektor współczynników jako

$$c = [c_1, c_2, ..., c_{2n}], \tag{26}$$

gdzie

$$c_j = \pi(\overline{x}_j) \cdot \{\pi(0 \mid \overline{x}_j, l) \cdot [CW01 + CW10] - CW10\} \tag{27}$$

możemy sformułować problem poszukiwania funkcji boolowskiej minimalizującej średni błąd bezwzględny jako Zadanie Programowania Liniowego.

$$\min_P c^T P \tag{28}$$

przy ograniczeniach

$$\overline{x}_j \le \overline{x}_i \Rightarrow P_B(1 \mid \overline{x}_j) \le P_B(1 \mid \overline{x}_i) \tag{29}$$

$$0 \le \overline{x}_j \le 1 \qquad i = 1,...,2^n \tag{30}$$

Pierwsza nierówność ograniczeń (29) wynika z konieczności spełniania przez funkcję boolowską stochastycznej właściwości układania w stos, druga (30) natomiast związana jest z dopuszczalnymi wartościami, jakie może przyjmować stochastyczna funkcja boolowska.

3. Rozwiązanie zadania programowania liniowego

Rozwiązanie sformułowanego powyżej ZPL zostanie podzielone na dwie części. W pierwszej obliczane będą współczynniki c_j na podstawie danych zawartych w przyjętych modelach sygnałów S (t) i N (t), a w drugiej ograniczenia wynikające z właściwości układania w stos i dopuszczalnych wartości stochastycznej funkcji boolowskiej zostaną zapisane w postaci nierówności macierzowych.

3.1. Obliczanie współczynników c_j

W celu obliczenia współczynników c_j rozpatrzymy, które parametry modeli sygnałów dane są na wstępie, a które musimy dopiero obliczyć. Parametry modeli sygnałów S (t) i N (t) są dane. Model sygnału R (t) znajdujemy na podstawie S (t) i N (t). Na podstawie R (t) dla danej szerokości apertury tworzymy model sygnału wektora aperturowego W (t). Następnie zostaną przedstawione wzory na obliczanie współczynników c_j na podstawie znalezionych modeli sygnałów.

3.1.1. Model S (t) i N (t)

Jako model sygnału S (t) przyjmujemy stacjonarny łańcuch Markowa z przestrzenią stanów $Q_S = \{S_1, ..., S_{M_S}\}$, gdzie M_S jest liczbą stanów przyjmowanych przez S (t) i macierzą przejść P_S.

Prawdopodobieństwa graniczne stanów $S_1, ..., S_{M_S}$ oznaczamy przez $h_1, ..., h_{M_S}$, gdzie

$$h_j = \lim_{t \to \infty} P(S(t) = S_j) \tag{31}$$

h_j obliczamy rozwiązując układ równań:

$$h_j = \sum_{j=1}^{M_S} h_i \cdot P_S(i,j), \qquad j = 1,...,M_S \tag{32}$$

$$\sum_{j=1}^{M_S} h_j = 1 \tag{33}$$

W celu numerycznego rozwiązania powyższego układu równań tworzymy macierz **A** i wektory **B** i **X**

$$\mathbf{A} = \begin{array}{c} \begin{array}{cc} 1 \ \dots \ M_S & \end{array} \\ \begin{bmatrix} \left[P_S^T - \mathbf{1} \right] \\ 1 \ \dots \ 1 \end{bmatrix} \end{array} \begin{array}{l} 1 \\ \vdots \\ M_S \\ M_S+1 \end{array}, \tag{34}$$

gdzie **1** jest macierzą jednostkową.

$$\mathbf{B} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \begin{array}{l} 1 \\ \vdots \\ M_S \\ M_S+1 \end{array} \qquad \mathbf{X} = \begin{bmatrix} h_1 \\ \vdots \\ h_{M_S} \end{bmatrix}, \tag{35}$$

Prawdopodobieństwa graniczne uzyskujemy jako rozwiązanie równania:

$$\mathbf{X} = \mathbf{A} \backslash \mathbf{B} \tag{36}$$

Jako model sygnału N (t) przyjmujemy stacjonarny łańcuch Markowa z przestrzenią stanów $Q_N = \{N_1, ..., N_{M_N}\}$, gdzie M_N jest liczbą stanów przyjmowanych przez N (t) i macierzą przejść P_N.

Prawdopodobieństwa graniczne stanów $N_1, ..., N_{M_N}$ oznaczamy przez $\upsilon_1, ..., \upsilon_{M_N}$, gdzie

$$\upsilon_j = \lim_{t \to \infty} P\,(N(t) = N_j) \tag{37}$$

υ_j obliczamy rozwiązując układ równań:

$$\upsilon_j = \sum_{j=1}^{M_N} \upsilon_i \cdot P_N(i,j), \qquad j = 1,...,M_N \tag{38}$$

$$\sum_{j=1}^{M_N} \upsilon_j = 1 \tag{39}$$

Obliczenia numeryczne υ_j przeprowadzamy analogicznie do obliczania h_j.

3.1.2. Model R (t)

Jako model sygnału R (t) przyjmujemy stacjonarny łańcuch Markowa z przestrzenią stanów $Q_R = \{r_1, ..., r_{M_R}\}$, gdzie M_R jest liczbą stanów przyjmowanych przez R (t) i macierzą przejść P_R.

$$M_R = M_S \cdot M_N \tag{40}$$

$$Q_R = Q_S \times Q_N \tag{41}$$

$$P_R = P_S \otimes P_N \tag{42}$$

Prawdopodobieństwo graniczne stanu $r_k = (S_i, N_j)$ przy założeniu niezależności S_i oraz N_j obliczamy jako:

$$g_k = \lim_{t \to \infty} P(R(t) = r_k) = h_j \cdot \upsilon_i \tag{43}$$

3.1.3. Model $\overline{W}(t)$

Jako model sygnału $\overline{W}(t)$ przyjmujemy stacjonarny łańcuch Markowa z przestrzenią stanów $Q_W = \{\overline{W}_1, ..., \overline{W}_{M_W}\}$, gdzie M_W jest liczbą stanów przyjmowanych przez $\overline{W}(t)$ i macierzą przejść P_W.

Sygnał $\overline{W}(t)$ powstaje przez nakładanie apertury o szerokości n na sygnał R (t) i przesuwaniu jej kolejno o jedną próbkę.

Oznacza to, że jeżeli $P_R(i,j) = 0$, czyli nie może nastąpić w sygnale R (t) przejście ze stanu r_i do stanu r_j, to do przestrzeni Q_W nie może należeć stan $\overline{W}_k$, w którym występowałyby jeden po drugim stany r_i i r_j. Zatem przestrzeń stanów Q_W możemy zdefiniować następująco:

$$\overline{W}_k = \left[r_{k_1}, r_{k_2}, ..., r_{k_n}\right] \in Q_W \Leftrightarrow P_R(k_i, k_{i+1}) \neq 0, i = 1,2,...,n-1 \tag{44}$$

Przestrzeń stanów Q_W generujemy według powyższego warunku. W wyniku tych obliczeń numerycznych uzyskujemy również M_W.

Obliczanie macierzy przejść P_W pokazane zostanie najpierw na przykładzie apertury o szerokości 3. Przejście ze stanu $\overline{W}_i$ do stanu $\overline{W}_j$ przebiega następująco:

$$\begin{aligned} \overline{W}_i &= \left[r_{i_1}, \underset{\downarrow}{r_{i_2}}, \underset{\downarrow}{r_{i_3}}\right] \\ \overline{W}_j &= \left[r_{j_1}, r_{j_2} \to r_{j_3}\right] \end{aligned} \tag{45}$$

Zatem prawdopodobieństwo przejścia ze stanu $\overline{W}_i$ do stanu $\overline{W}_j$ wynosi:

$$P_W(i,j) = P(r_{i_2} = r_{j_1}) \cdot P(r_{i_3} = r_{j_2}) \cdot P(r_{j_2} \to r_{j_3}) \tag{46}$$

$$P_W(i,j) = P(r_{i_2} = r_{j_1}) \cdot P(r_{i_3} = r_{j_2}) \cdot P_R(j_2, j_3), \tag{47}$$

gdzie $P(r_{i2} = r_{j1})$ oznacza prawdopodobieństwo, że stan r_{i2} jest tym samym stanem co r_{j1} (przyjmujemy 0, jak nie jest, a 1 - gdy jest). $P(r_{j2} \rightarrow r_{j3})$ oznacza prawdopodobieństwo przejścia ze stanu r_{j2} do stanu r_{j3}.

Dla szerokości apertury n obliczamy kolejne $P_W(i,j)$, $i = 1, ..., M_W$ jako

$$P_W(i,j) = P(r_{i_2} = r_{j_1}) \cdot P(r_{i_3} = r_{j_2}) \cdot ... \cdot P(r_{i_n} = r_{j_{n-1}}) \cdot P_R(j_{n-1}, j_n) \tag{48}$$

Przez $\pi(\overline{W}_k)$, $k = 1, ..., M_W$ oznaczymy prawdopodobieństwa graniczne

$$\pi(\overline{W}_k) = \lim_{t \rightarrow \infty} P(\overline{W}(t) = \overline{W}_k) \tag{49}$$

$\pi(\overline{W}_k)$ obliczamy rozwiązując układ równań:

$$\pi(\overline{W}_j) = \sum_{j=1}^{M_W} \pi(\overline{W}_i) \cdot P_W(i,j), \quad j = 1,...,M_W \tag{50}$$

$$\sum_{j=1}^{M_W} \pi(\overline{W}_j) = 1 \tag{51}$$

Obliczenia numeryczne wykonujemy tak samo, jak pokazano dla S(t).

3.1.4. Obliczanie c_j

Współczynnik c_j obliczamy z wyprowadzonej uprzednio zależności

$$c_j = \pi(\overline{x}_j) \cdot \{\pi(0 \mid \overline{x}_j, 1) \cdot [CW01 + CW10] - CW10\}, \tag{52}$$

gdzie

$$\pi(\overline{x}_j) = \sum_{\overline{W}_k \in Q_W : \overline{w}_{k,1=1} = \overline{x}_j} \pi(\overline{W}_k) \tag{53}$$

Sposób obliczania prawdopodobieństw granicznych $\pi(\overline{W}_k)$ pokazano w podrozdziale 3.1.3.

Prawdopodobieństwo warunkowe pojawienia się zera w S (t) i pojawienia się na wejściu filtru $\overline{x}_j$ obliczamy jako

$$\pi(0 \mid \overline{x}_j) = \frac{\pi(0, \overline{x}_j)}{\pi(\overline{x}_j)}, \tag{54}$$

gdzie

$$\pi(0,\overline{x}_j) = \sum_{\overline{W}_k \in Q_w : \left[(\overline{w}_{k,l=1} = \overline{x}_j) \& \left(\left[S \approx r_{k_{n/2}} \right] = 0 \right) \right]} \pi(\overline{W}_k) \tag{55}$$

Warunek $[S \sim r_{k_{n/2}}] = 0$ należy rozumieć następująco:

$$\overline{W}_k = \left[r_{k_1}, \cdots, r_{k_{n/2}}, \cdots, r_{k_n} \right]$$

$$r_{k_{n/z2}} = (S_m, N_p)$$

jeżeli $S_m = 0$, to $\overline{W}_k$ spełnia warunek.

Jeżeli natomiast przyjmiemy koszty CW01 = CW10 = 1, to minimalizacja otrzymanego w ten sposób wskaźnika jest równoważna minimalizacji średniego błędu bezwzględnego

$$C(S_B) = E[\,|\,S(t) - S_B(\overline{W}(t))\,|\,] \tag{56}$$

3.2. Przeformułowanie ograniczeń wynikających z właściwości układania w stos do postaci ograniczeń nierównościowych ZPL

Rozważając ograniczenia, które musi spełniać funkcja boolowska, aby mogła być użyta do stworzenia filtru stosowego, chcielibyśmy rozwiązać dwa związane z tymi ograniczeniami problemy. Właściwość układania w stos narzuca konieczność spełniania warunku $\overline{x}_i \le \overline{x}_j \Rightarrow B(\overline{x}_i) \le B(\overline{x}_j)$. A zatem z jednej strony chcielibyśmy zminimalizować liczbę porównań koniecznych do sprawdzenia, czy powyższy warunek zawsze zachodzi, a z drugiej strony chcielibyśmy zapisać te porównania w postaci nierówności macierzowej, którą można by wykorzystać jako nierówność ograniczeń ZPL.

Wprowadźmy następujące oznaczenia:

(r0s) - ciąg n binarnych zmiennych, r - podciąg o długości l_r, s - podciąg o długości l_s, $l_r + l_s + l = n$

(r0s,r1s) - para dwóch ciągów binarnych różniących się na jednej pozycji

P_B (r0s,1) - prawdopodobieństwo zdarzenia: B(r0s) = 1

Definicja lokalnej właściwości układania w stos

O funkcji boolowskiej B (·) mówimy, że posiada lokalnie (dla danej pary (r0s, r1s)) właściwość układania w stos, jeżeli

$$P_B(r0s,1) \le P_B(r1s,1) \tag{57}$$

Funkcja boolowska B (·) posiada właściwość układania w stos, jeżeli posiada lokalną właściwość układania w stos dla wszystkich możliwych par (r0s,r1s).

Spróbujmy zatem zapisać w postaci nierówności macierzowej warunek posiadania lokalnej właściwości układania w stos dla wszystkich możliwych par (r0s, r1s).

Przyjmijmy, że ustaliliśmy pewną kolejność w zbiorze par (r0s, r1s). Zatem dla pary o numerze q możemy zapisać:

$$P_B(r0s,1) \le P_B(r1s,1) \quad \text{dla pary } q \tag{58}$$

Przechodząc na oznaczenia związane z definicją stochastycznej funkcji boolowskiej i przyjmując, że $\overline{x}_i = r0s$ oraz $\overline{x}_j = r1s$, możemy zapisać powyższą nierówność jako:

$$P_B(1 \mid \overline{x}_i) \le P_B(1 \mid \overline{x}_j) \quad \text{dla pary } q, \tag{59}$$

zatem

$$P_i \le P_j \quad \text{dla pary } q \tag{60}$$

$$P_i - P_j \le 0 \text{ dla pary } q \tag{61}$$

Przyjmując, że numer kolumny macierzy odpowiada numerowi $\overline{x}_k$, $k = 1,\ldots,2^n$, a numer wiersza numerowi pary (r0s,r1s), przedstawimy ideę tworzenia macierzy ograniczeń w następujący sposób:

$$\begin{array}{c} 1 \\ \vdots \\ q \\ \vdots \\ n2^{n-1} \end{array} \begin{array}{c} 1 \cdots i \cdots j \cdots 2^n \\ \left[\begin{array}{ccccccc} & & & & & & \\ & & & & & & \\ 0\cdots0 & 1 & 0\cdots0 & -1 & 0\cdots0 \\ & & & & & & \end{array}\right] \end{array} \begin{bmatrix} P_1 \\ \vdots \\ P_i \\ \vdots \\ P_j \\ \vdots \\ P_{2^n} \end{bmatrix} \le \begin{array}{c} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} \end{array} \begin{array}{c} 1 \\ \vdots \\ q \\ \vdots \\ n2^{n-1} \end{array}$$

Z przedstawionego schematu mnożenia macierzy wynika sposób zapisu w postaci nierówności macierzowej warunku: $P_i - P_j \le 0$ dla pary q. Oznaczając natomiast poszczególne macierze składowe jako:

$$A_l = \begin{array}{c} \\ 1 \\ \vdots \\ q \\ \vdots \\ n2^{n-1} \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 1 & \cdots & i & \cdots & j & \cdots & 2^n \end{array} \\ \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & -1 & 0 & \cdots & 0 \end{bmatrix} \end{array} \tag{62}$$

$$P^T = \begin{bmatrix} P_1 \\ \vdots \\ P_i \\ \vdots \\ P_j \\ \vdots \\ P_{2^n} \end{bmatrix} \quad b_l = \begin{array}{c} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} \end{array} \begin{array}{c} 1 \\ \vdots \\ q \\ \vdots \\ n2^{n-1} \end{array} \tag{63}$$

możemy zapisać ograniczenia zapewniające posiadanie przez uzyskaną funkcję boolowską właściwości układania w stos w następującej postaci:

$$A_l P^T \le b_l \tag{64}$$

Pozostaje jeszcze uwzględnić w macierzy ograniczeń dopuszczalny zakres wartości dla P_i. Mianowicie

$$0 \le P_i \le 1 \tag{65}$$

Jeżeli przyjmiemy $\overline{x}_1 = [0...0]$ oraz $\overline{x}_{2^n} = [1...1]$, to dzięki właściwości układania w stos warunek $(0 \le P_i \le 1)$ jest spełniony, gdy zachodzi $P_1 \ge 0$ i $P_{2^n} \le 1$, a to zapewnia nam następująca modyfikacja macierzy A_l i wektora b_l:

$$A = \begin{array}{c} \begin{array}{ccc} 1 & \cdots & 2^n \end{array} \\ \begin{bmatrix} -1 & 0 & \cdots & 0 \\ & & A_l & \\ 1 & \cdots & 0 & 1 \end{bmatrix} \end{array} \begin{array}{c} 1 \\ \vdots \\ n2^{n-1}+2 \end{array} \tag{66}$$

$$b = \begin{bmatrix} 0 \\ \begin{bmatrix} b_l \end{bmatrix} \\ 1 \end{bmatrix} \begin{array}{c} 1 \\ \vdots \\ n2^{n-1}+2 \end{array} \tag{67}$$

Reasumując, stwierdzamy, że funkcję boolowską minimalizującą średni błąd bezwzględny otrzymujemy w wyniku rozwiązania następującego ZPL:

$$\min_{P} c^T P$$
$$AP^T \leq b, \tag{68}$$

gdzie A i b zostały wyprowadzone powyżej; sposób obliczania c zawiera podrozdział 3.1, a P jest wektorem prawdopodobieństw określającym funkcję boolowską. Zauważmy, że ze względu na właściwości rozwiązania ZPL (znajduje się ono zawsze na ograniczeniach) prawdopodobieństwa P_i są zawsze równe 0 lub 1. Zatem otrzymana funkcja boolowska jest deterministyczną funkcją boolowską.

4. Przykłady optymalnych funkcji boolowskich

Podobnie jak w [7] przyjęto następujący model:

S (t) jest dyskretnym w czasie łańcuchem Markowa z przestrzenią stanów

$$Q_S = \{(0,a), (0,b), (1,a), (1,b)\} \tag{69}$$

i macierzą przejść

$$P_S = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.4 & 0.6 & 0 \\ 0 & 0 & 0 & 1 \\ 0.4 & 0 & 0 & 0.6 \end{bmatrix} \tag{70}$$

$M_S = 4$

S (t) jest sygnałem binarnym. Wyróżniane są dwa stany, których wystąpienie równoznaczne jest z pojawieniem się S (t) = 0. Są to stany (0,a) i (0,b). Jednocześnie występują dwa stany dające S (t) = 1. Są to stany (1,a) i (1,b). Analizując P_S widzimy, że jeżeli sygnał znajduje się w stanie „0" lub „1", to pozostaje w nim co najmniej przez jeden dodatkowy przedział czasowy. Innymi słowy, jeżeli przełączenie z „0" na „1" dało nam w chwili k S(t) = 1, to również w k + 1 będzie S (t) = 1, a w chwili k + 2 będzie 1 z prawdopodobieństwem 0.6, a 0 z prawdopodobieństwem 0.4. Stany (0,a), (0,b) są nierozróżnialne, podobnie jak (1,a), (1,b).

Funkcja $f_S((0,a)) : Q_S \rightarrow \{0,1\}$, która definiuje wartości sygnału S (t), ma następującą postać:

$$f_S((x,y)) = x \tag{71}$$

Zatem $f_S((0,a)) = 0$ a $f_S((1,b)) = 1$.

Rozwiązując układ równań (32), (33) uzyskujemy prawdopodobieństwa graniczne dla stanów z przestrzeni Q_S

$$h_1 = 0.1622 \quad h_2 = 0.2703 \quad h_3 = 0.1622 \quad h_4 = 0.4054 \tag{72}$$

N (t) jest dyskretnym w czasie łańcuchem Markowa z przestrzenią stanów

$$Q_N = \{X, Y, Z\} \tag{73}$$

i macierzą przejść

$$P_N = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.8 & 0.1 & 0.1 \\ 0.6 & 0.3 & 0.1 \end{bmatrix} \tag{74}$$

$M_N = 3$

Zakłada się, że szum jest niezależny od sygnału idealnego.

Stany X,Y,Z oznaczają następujący charakter oddziaływań zakłócenia na sygnał idealny:

X - N (t) nie wpływa na S (t)

Y - N (t) sprowadza S (t) do 1

Z - N (t) sprowadza S (t) do 0

Rozwiązując układ równań (38), (39) uyskujemy prawdopodobieństwa graniczne dla stanów z przestrzeni Q_N

$$\nu_1 = 0.6393 \quad \nu_2 = 0.1967 \quad \nu_3 = 0.1639 \tag{75}$$

R (t) jest dyskretnym w czasie łańcuchem Markowa z przestrzenią stanów

$$\begin{aligned} Q_R = Q_S \times Q_N = \{&(0,a,X), (0,a,Y), (0,a,Z), (0,b,X), (0,b,Y), (0,b,Z), \\ &(1,a,X), (1,a,Y), (1,a,Z), (1,b,X), (1,b,Y), (1,b,Z)\} \end{aligned} \tag{76}$$

Funkcja $f_R(r) : Q_R \rightarrow \{0,1\}$, która definuje sygnał R (t), w chwili t ma następującą postać:

$f_R(r) \in \{0,1\}$, dla każdego $r \in Q_R$, tak więc $f_R(0,a,X) = 0$, podczas gdy $f_R(0,a,Y) = 1$.

Macierz przejść ma postać:

$$P_R = \begin{bmatrix} 0 & P_N & 0 & 0 \\ 0 & 0.4P_N & 0.6P_N & 0 \\ 0 & 0 & 0 & P_N \\ 0.4P_N & 0 & 0 & 0.6P_N \end{bmatrix} \tag{77}$$

P_R jest macierza 12 na 12 elementów.

Korzystając z (43) otrzymujemy prawdopodobieństwa graniczne g_k według kolejności stanów w Q_R.

$$g_1 = 0.1037, \quad g_2 = 0.0319, \quad g_3 = 0.0266,$$
$$g_4 = 0.1728, \quad g_5 = 0.0532, \quad g_6 = 0.0443,$$
$$g_7 = 0.1037, \quad g_8 = 0.0319, \quad g_9 = 0.0266,$$
$$g_{10} = 0.2592, \quad g_{11} = 0.0798, \quad g_{12} = 0.0665$$

$M_R = 12$

Parametry sygnału W (t), tj. przestrzeń stanów Q_W, macierz przejść P_W i prawdopodobieństwa graniczne $\pi(\overline{W}_j)$ znajdujemy według opisu zamieszczonego w rozdziale 3.1.3.

Zadanie optymalizacji zgodnie z (68) ma postać:

$$\min_P c^T P \tag{78}$$

$$AP^T \le b \tag{79}$$

Sposób wyliczania współczynników c_j przedstawiono w rozdziale 3.1. Przyjmując, że szerokość apertury filtru wynosi 3, otrzymujemy następujące postacie macierzy ograniczeń A i b.

$$A = \begin{matrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{matrix} \qquad b = \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{matrix}$$

Kolejność kombinacji zmiennych wejściowych $\overline{x}_j$ pokazano w tab.4.1.

Tabela 4.1

j	$\bar{x}_j$
1	0 0 0
2	1 0 0
3	0 1 0
4	0 0 1
5	1 1 0
6	1 0 1
7	0 1 1
8	1 1 1

W tabelach od 4.2. do 4.21 zebrano przykładowe wyniki uzyskane dla różnych wartości współczynników kosztów CW01, CW10. Dla CW01 = CW10 = 1 otrzymana funkcja boolowska minimalizuje średni błąd bezwzględny. Dla współczynników kosztu różnych od 1 uzyskujemy minimalizację ważonego średniego błędu bezwzględnego. Analizując wpływ wartości CW01, CW10 na postać otrzymywanej funkcji widzimy, że w miarę jak wzrasta kara za ustawienie przez filtr na wyjściu 1, gdy powinno być 0 (odpowiada temu wzrost CW01), filtr stara się „zabezpieczyć" przed taką sytuacją przyjmując dla coraz większej liczby kombinacji zmiennych wejściowych 0. Przeciwnie, jeżeli zwiększamy karę za wyjście filtru równe 0, gdy powinno być 1.

Tabela 4.2

CW01	CW10	C(B)
0.01	1	0.0043
c_j	$P_B(1 \mid \bar{x}_j)$	j
-0.0037	1	1
-0.0196	1	2
-0.0441	1	3
-0.0244	1	4
-0.1268	1	5
-0.0418	1	6
-0.1220	1	7
-0.1807	1	8

Tabela 4.3

CW01	CW10	C(B)
0.1	1	0.0347
c_j	$P_B(1 \mid \bar{x}_j)$	j
0.0039	0	1
-0.0099	1	2
-0.0417	1	3
-0.0152	1	4
-0.1247	1	5
-0.0371	1	6
-0.1194	1	7
-0.1802	1	8

Tabele 4.2 do 4.5 przedstawiają wartości minimalizowanego wskaźnika C (B), współczynników c_j oraz tabele prawdy funkcji boolowskiej uzyskanych dla różnych CW01, CW10 w przykładzie 1.

Tabela 4.4

CW01	CW10	C(B)
0.02	1	0.0480
c_j	$P_B(1 \mid \bar{x}_j)$	j
0.0124	0	1
0.0008	0	2
-0.0390	1	3
-0.0049	1	4
-0.1223	1	5
-0.0319	1	6
-0.1165	1	7
-0.1797	1	8

Tabela 4.5

CW01	CW10	C(B)
0.7	1	0.0960
c_j	$P_B(1 \mid \bar{x}_j)$	j
0.0550	0	1
0.0546	0	2
-0.0253	1	3
0.0464	0	4
-0.1104	1	5
-0.0059	1	6
-0.1023	1	7
-0.1770	1	8

Tabela 4.6

CW01	CW10	C(B)
1	1	0.0851
c_j	$P_B(1 \mid \bar{x}_j)$	j
0.0806	0	1
0.0868	0	2
-0.0171	1	3
0.0772	0	4
-0.1033	1	5
0.0097	0	6
-0.0937	1	7
-0.1753	1	8

Tabela 4.7

CW01	CW10	C(B)
2	1	0.1156
c_j	$P_B(1 \mid \bar{x}_j)$	j
0.1658	0	1
0.1942	0	2
0.0101	0	3
0.1799	0	4
-0.0795	1	5
0.0618	0	6
-0.0652	1	7
-0.1698	1	8

Tabela 4.8

CW01	CW10	C(B)
5	1	0.1462
c_j	$P_B(1 \mid \bar{x}_j)$	j
0.4214	0	1
0.5166	0	2
0.0920	0	3
0.4879	0	4
-0.0082	1	5
0.2179	0	6
0.0205	0	7
-0.1534	1	8

Tabela 4.9

CW01	CW10	C(B)
6	1	0.0329
c_j	$P_B(1 \mid \bar{x}_j)$	j
0.5066	0	1
0.6241	0	2
0.1192	0	3
0.5906	0	4
0.0156	0	5
0.2700	0	6
0.0490	0	7
-0.1479	1	8

Tabela 4.10

CW01	CW10	C(B)
30	1	0.1643
c_j	$P_B(1 \mid \bar{x}_j)$	j
2.5512	0	1
3.2029	0	2
0.7738	0	3
3.0546	0	4
0.5859	0	5
1.5192	0	6
0.7343	0	7
0.0165	1	8

Tabela 4.11

CW01	CW10	C(B)
35	1	0.0000
c_j	$P_B(1 \mid \bar{x}_j)$	j
2.9771	0	1
3.7402	0	2
0.9102	0	3
3.5679	0	4
0.7048	0	5
1.7795	0	6
0.8770	0	7
0.0109	0	8

Tabela 4.12

CW01	CW10	C(B)
1	0.01	0.0000
c_j	$P_B(1 \mid \bar{x}_j)$	j
0.0851	0	1
0.1072	0	2
0.0268	0	3
0.1024	0	4
0.0225	0	5
0.0516	0	6
0.0273	0	7
0.0037	0	8

Tabela 4.13

CW01	CW10	C(B)
1	0.05	0.0055
c_j	$P_B(1 \mid \bar{x}_j)$	j
0.0850	0	1
0.1064	0	2
0.0251	0	3
0.1014	0	4
0.0174	0	5
0.0499	0	6
0.0224	0	7
-0.0036	1	8

Tabele 4.10 do 4.13 przedstawiają wartości minimalizowanego wskaźnika C (B), współczynników c_j oraz tabele prawdy funkcji boolowskich uzyskiwanych dla różnych CW01, CW10 w przykładzie 1.

Tabela 4.14

CW01	CW10	C(B)
1	0.1	0.0055
c_j	$P_B(1 \mid \overline{x}_j)$	j
0.0847	0	1
0.1054	0	2
0.0228	0	3
0.1001	0	4
0.0111	0	5
0.0478	0	6
0.0163	0	7
-0.0126	1	8

Tabela 4.15

CW01	CW10	C(B)
1	0.5	0.0578
c_j	$P_B(1 \mid \overline{x}_j)$	j
0.0829	0	1
0.0971	0	2
0.0051	0	3
0.0899	0	4
-0.0398	1	5
0.0309	0	6
-0.0326	1	7
-0.0849	1	8

Tabela 4.16

CW01	CW10	C(B)
1	0.8	0.0851
c_j	$P_B(1 \mid \overline{x}_j)$	j
0.0815	0	1
0.0909	0	2
-0.0083	1	3
0.0823	0	4
-0.0779	1	5
0.0182	0	6
-0.0693	1	7
-0.1392	1	8

Tabela 4.17

CW01	CW10	C(B)
1	1	0.0851
c_j	$P_B(1 \mid \overline{x}_j)$	j
0.0806	0	1
0.0868	0	2
-0.0171	1	3
0.0772	0	4
-0.1033	1	5
0.0097	0	6
-0.0937	1	7
-0.1753	1	8

Tabele 4.14 do 4.17 przedstawiają wartości minimalizowanego wskaźnika C (B), współczynników c_j oraz tabele prawdy funkcji boolowskich uzyskiwanych dla różnych CW01, CW10 w przykładzie 1.

Tabela 4.18

CW01	CW10	C(B)
1	2	0.1371
c_j	$P_B(1 \mid \bar{x}_j)$	j
0.0760	0	1
0.0661	0	2
-0.0616	1	3
0.0518	0	4
-0.2303	1	5
-0.0326	1	6
-0.2160	1	7
-0.3561	1	8

Tabela 4.19

CW01	CW10	C(B)
1	5	0.2398
c_j	$P_B(1 \mid \bar{x}_j)$	j
0.0622	0	1
0.0042	0	2
-0.1948	1	3
-0.0245	1	4
-0.6115	1	5
-0.1597	1	6
-0.5827	1	7
-0.8985	1	8

Tabela 4.20

CW01	CW10	C(B)
1	6	0.3472
c_j	$P_B(1 \mid \bar{x}_j)$	j
0.0576	0	1
-0.0165	1	2
-0.2393	1	3
-0.0500	1	4
-0.7385	1	5
-0.2020	1	6
-0.7050	1	7
-1.0793	1	8

Tabela 4.21

CW01	CW10	C(B)
1	20	0.4324
c_j	$P_B(1 \mid \bar{x}_j)$	j
-0.0068	1	1
-0.3057	1	2
-0.8612	1	3
-0.4061	1	4
-2.5171	1	5
-0.7949	1	6
-2.4166	1	7
-3.6105	1	8

Tabele 4.18 do 4.21 przedstawiają wartości minimalizowanego wskaźnika C (B), współczynników c_j oraz tabele prawdy funkcji boolowskich uzyskiwanych dla różnych CW01, CW10 w przykładzie 1.

5. Zakończenie

W pracy przedstawiono podstawowe definicje i właściwości przetwarzania stosowego, zaliczającego się do nieliniowych przetworników sygnałów, a będącego liniową kombinacją statystyk porządkowych. Następnie sformułowano zadanie filtracji optymalnej z uwagi na

średni błąd bezwzględny jako zadanie programowania liniowego. Przedstawiono metodę znajdowania współczynników liniowej funkcji wskaźnika jakości i omówiono warunki, jakie musi spełniać macierzowa reprezentacja właściwości układania w stos. Następnie przedstawiono przykłady optymalnych funkcji boolowskich.

LITERATURA

1. Coyle Edward, Gallagher Neal, Wendt Peter (August 1986) Stack Filters, IEEE Transactions On Acoustics, Speech And Signal Processing, vol. ASS-34, pp.898-911.
2. Coyle Edward, Lin Jean-Hsang (August 1988) Stack Filters and the Mean Absolute Error Criterion, IEEE Transactions On Acoustics, Speech And Signal Processing, vol. 36, No. 8.
3. Kuś Z., Wojciechowski K.: Przetwarzanie stosowe. Zeszyty Naukowe Politechniki Śląskiej ser. Automatyka, z.113, Gliwice 1994
4. Kuś Z., Wojciechowski K.: Deterministyczne właściwości przetwarzania stosowego. Zeszyty Naukowe Politechniki Śląskiej, ser. Automatyka, z.113, Gliwice 1994.
5. Kuś Z., Wojciechowski K.: Statystyczne właściwości przetwarzania stosowego. Zeszyty Naukowe Politechniki Śląskiej, ser. Automatyka, z.113, Gliwice 1994.
6. Kuś Z., Wojciechowski K.: Przetwarzanie medianowe jako szczególny przypadek przetwarzania stosowego. Zeszyty Naukowe Politechniki Śląskiej, ser. Automatyka, z.113, Gliwice 1994.
7. M. Gabbouj, (December 1986) Optimal stack filter examples and positive Boolean functions, M.S. thesis, School Elec. Eng. Purdue Univ., West Lafayette, In.

Recenzent: Prof.zw.dr hab.inż. Wojciech Zamojski

Wpłynęło do Redakcji 7.12.1995 r.

Abstract

The class of stack filters preserving stacking property and threshold decomposition is introduced. Rank order filters (especially median filters), morphological filters and many others are included in this class.

Basic features of stack filtering are presented.

The stack filter is fully determined by selection of appropriate Boolean function. As a criteria for selection of the function the minimization of the mean absolute error between required signal and the filter output is taken. The minimization of mean absolute error can be obtained through solution of the appropriate linear programming problem.

The subject of this paper is to present some of numerical problems arising during formulation of linear programming problem.

These enhanced methods allow for computation of the coefficients and the matrix of constraints for the linear programming problem formulated above.

Finally, examples of stack filters are presented.