POLSKA AKADEMIA NAUK
KOMITET ELEKTRONIKI I TELEKOMUNIKACJI

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI

ELECTRONICS AND TELECOMMUNICATIONS QUARTERLY

TOM 46 — ZESZYT 2

WYDAWNICTWO NAUKOWE PWN

WARSZAWA 2000

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. 11,0. Ark. druk. 9,5	Podpisano do druku w lipcu 2000 r.
Papier offset. kl. III 70 g. B-1	Druk ukończono w sierpniu 2000 r.

SKŁAD I ŁAMANIE: Agnieszka Chmielewska, Warszawa, ul. Korytnicka 28
DRUK I OPRAWA: Drukarnia Braci Grodzickich, Piaseczno, ul. Geodetów 47a

Szanowni Autorzy

„Kwartalnik Elektroniki i Telekomunikacji" — Electronics and Telecommunications Quartery jest kontynuatorem tradycji powstałego 46 lat temu kwartalnika p.t. „Rozprawy Elektrotechinczne".

Kwartalnik jest czasopismem Komitetu Elektroniki i Telekomunikacji Polskiej Akademii Nauk. Wydawany jest przez Wydawnictwo Naukowe PWN SA w Warszawie. Kwartalinik 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 współczasnej 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 rozowojowych. 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 specjalistów, 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 zagraniczych. 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

M

wych
staw
linio
przez
Turb
Win
linio
oper
prog

Slow
tura
czyn
Wie

Artyk
i paramet
J. Olędzk
teoria wy
jak i niel
autora a
Turbo W

* Pracę
Badań Nau

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2000, 46,z. 2, ss. 129–147

Modelowanie liniowych systemów stacjonarnych za pomocą systemów Laguerre'a*

GRZEGORZ CIESIELSKI

Instytut Elektroniki Teoretycznej, Metrologii i Materiałoznawstwa, Politechnika Łódzka

Otrzymano 1999.05.10
Autoryzowano 2000.02.24

Artykuł omawia zagadnienia identyfikacji strukturalnej i parametrycznej modeli liniowych systemów stacjonarnych. Do opisu przyjęto strukturę modeli Laguerre'a. Przedstawiona tu teoria wynika z jednolitej ogólnej teorii operatorowej systemów, zarówno liniowych, jak i nieliniowych. Jednolita ogólna teoria operatorowa systemów jest rozwijana przez autorów artykułu od szeregu lat ([7]–[24]). W artykule omówiono również program Turbo Wiener ™, opracowany i rozwijany przez autora dla systemu operacyjnego MS Windows, który pozwala wyznaczyć modele i korektory systemów stacjonarnych, zarówno liniowych, jak i nieliniowych, wykorzystując przy tym wspomnianą jednolitą ogólną teorię operatorową systemów. Przedstawiono tu również wyniki obliczeń otrzymanych za pomocą programu Turbo Wiener™, dla typowych liniowych systemów stacjonarnych.

Słowa kluczowe: ogólna funkcjonalna teoria systemów, modelowanie, identyfikacja strukturalna, identyfikacja parametryczna, systemy liniowe, systemy stacjonarne, systemy przyczynowe, operatory, sploty, przestrzeń splotowa, przestrzeń przyczynowa, program Turbo Wiener™

1. WSTĘP

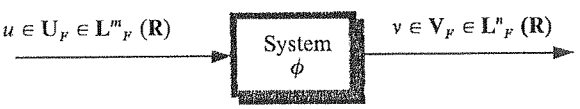
Artykuł omawia zagadnienia modelowania, a więc identyfikację strukturalną i parametryczną modeli liniowych systemów stacjonarnych (J. Jworski, R. Morawski, J. Olędzki [31]). Do ich opisu przyjęto strukturę modeli Laguerre'a. Przedstawiona tu teoria wynika z jednolitej ogólnej teorii operatorowej systemów, zarówno liniowych, jak i nieliniowych. Jednolita ogólna teoria operatorowa systemów rozwijana przez autora artykułu od szeregu lat ([7]–[24]). W artykule omówiono również program Turbo Wiener™, opracowany i rozwijany przez autora dla systemu operacyjnego MS

* Pracę wykonano w ramach projektu badawczego nr 8 T10C 031 09 finansowanego przez Komitet Badań Naukowych w latach 1995–1997.

Windows. Program ten pozwala wyznaczyć modele i korektory systemów stacjonarnych, zarówno liniowych, jak i nieliniowych. Wykorzystuje on przy tym wspomianą jednolitą ogólną teorię operatorową systemów. Program Turbo Wiener™ został w niniejszym artykule omówiony jedynie w części dotyczącej modelowania liniowych systemów stacjonarnych. Przedstawiono tu również wyniki obliczeń otrzymanych za pomocą programu Torbo Wiener™ , dla typowych liniowych systemów stacjonarnych, inercyjnego i oscylacyjnego. W artykule omówiono teorię liniowych systemów stacjonarnych przyczynowych z jednym wejściem i jednym wyjściem. Wynika to z fakty, że dla systemów wielowymiarowych niezbędnym jest wprowadzenie dość złożonego pojęcia iloczynu tensorowego przestrzeni liniowych, co wykracza poza ramy tego artykułu, a co będzie przedstawione w jednym z następnych artykułów.

2. PRELIMINARIA MATEMATYCZNE

Dowolny system można traktować jako odwzorowanie zbioru U sygnałów wejściowych zwanych *wymuszeniami* w zbiór V sygnałów wyjściowych zwanych *odpowiedziami*. Będziemy dalej zakładać, że oba te zbiory wyposażone są w strukturę przestrzeni liniowej, a zatem określona jest w nich operacja dodawania sygnałów i mnożenia sygnału przez liczbę z ciała liczbowego $\mathbf{F}$[1].



Rys. 2. 1. Podstawowy model systemu.

Niech R[2] będzie zbiorem chwil czasowych, dla których określone są sygnały z U oraz V. Załóżmy ponadto, że $m \in \mathbf{N}$[3] jest liczbą wejść, a $n \in \mathbf{N}$ liczbą wyjść systemu. Wartości sygnałów wejściowych należą zatem do $\mathbf{F}^m$[4], a wartości sygnałów wyjściowych do $\mathbf{F}^n$. Wówczas przestrzeń sygnałów wejściowych $\mathbf{U}_F$[5] jest podprzestrzenią liniową przestrzeni $\mathbf{L}^m{}_F(\mathbf{R})$[6]. Podobnie, prestrzeń sygnałów wyjściowych $\mathbf{V}_F$ jest podprzestrzenią liniową przestrzeni $\mathbf{L}^n{}_F(\mathbf{R})$, co zostało pokazane na rys. 2.1. Ponieważ $\mathbf{U}$ oraz $\mathbf{V}$ są przestrzeniami funkcyjnymi, których elementy określone są na samej dziedzinie $\mathbf{R}$, a dowolne odwzortowanie ϕ takich przestrzeni nazywane jest

[1] W całym artykule przez $\mathbf{F}$ oznaczamy ciało liczbowe. W szczególności ciało $\mathbf{F}$ oznacza ciało liczb rzeczywistych $\mathbf{R}$ lub ciało liczb zespolonych $\mathbf{C}$.

[2] Zbiór X wyposażony w dowolną strukturę algebraiczną i/lub topologiczną będziemy oznaczać literą X, a zatem przez R oznaczamy zbiór liczb rzeczywistych, natomiast przez $\mathbf{R}$ ciało liczb rzeczywistych.

[3] Przez N oznaczamy zbiór liczb naturalnych.

[4] Dla dowolnego zbioru X przez X^n oznaczamy n-krotny iloczyn kartezjański (produkt) zbioru X.

[5] Przez $\mathbf{X}_F$ oznaczamy przestrzeń liniową nad ciałem skalarów $\mathbf{F}$. Jeżeli $\mathbf{F}$ wynika z kontekstu, to jest ono pomijane w jej oznaczeniu.

[6] Przez $\mathbf{L}^n_F(X)$ oznaczamy przestrzeń odwzorowań określonych na X o wartościach z $\mathbf{F}^n$. Oznaczenia $\mathbf{L}^1_F(X)$ oraz $\mathbf{L}_F(X)$ traktujemy jako równoważne, a gdy zbiór X wynika z kontekstu, to piszemy w skrócie $\mathbf{L}^n_F$ lub L_F.

operatorem, to każdy system można opisać za pomocą operatora, dla którego jedną z możliwych form reprezentacji jest układ równań różniczkowych. By uprościć dalszą dyskusję, pojęcia: *system opisany za pomocą operatora* ϕ *i operator* φ, będą więc traktowane jako równoważne.

W modelowaniu systemów wykorzystuje się trzy elementarne operatory: przesunięcie, obtót i splotowe przesunięcie funkcyjne (G. Ciesielski [11]–[18] oraz [20]–[24]).

Przesunięciem funkcyjnym o $\sigma \in \mathbf{R}^k$ nazywamy operator oznaczany przez $\nabla_\sigma : \mathbf{L}^m{}_F(\mathbf{R}^k) \mapsto \mathbf{L}^m{}_F(\mathbf{R}^k)$, który $\forall u \in \mathbf{L}^m{}_F(\mathbf{R}^k)$ i $\forall t \in \mathbf{R}^k$ spełnia warunek:

$$[\nabla_\sigma(u)](t) := u(t+\sigma)^{7}.$$

Funkcję $\nabla_\sigma(u)$ nazywanym *przesunięciem funkcji* u o σ.

Obrotem funkcyjnym nazywamy operator, który oznaczamy przez $\nabla^* : \mathbf{L}^m{}_F(\mathbf{R}^k) \mapsto \mathbf{L}^m{}_F(\mathbf{R}^k)$ taki, że $\forall u \in \mathbf{L}^m{}_F(\mathbf{R}^k)$ i $\forall t \in \mathbf{R}^k$

$$[\nabla^*(u)](t) := u(-t).$$

Funkcję $\nabla^*(u)$ nazywamy *obrotem funkcji* u.

Splotowym przesunięciem funkcyjnym o $\sigma \in \mathbf{R}^k$ nazywamy operator, który oznaczany jest przez $\nabla^*_\sigma : \mathbf{L}^m_F(\mathbf{R}^k) \to \mathbf{L}^m_F(\mathbf{R}^k)$ oraz $\forall u \in \mathbf{L}^m_F(\mathbf{R}^k)$ i $\forall t \in \mathbf{R}^k$

$$[\nabla^*_\sigma(u)](t) := u(\sigma - t).$$

Funkcję $\nabla^*_\sigma(u)$ nazywać będziemy *przesunięciem splotowym funkcji* u o σ.

Zauważym tu, że przesunięcia ∇_σ i ∇^*_σ są operatorami liniowymi na przestrzeni $\mathbf{L}^m{}_F(\mathbf{R}^k)$ i odwzorowują przestrzeń $\mathbf{L}^m{}_F(\mathbf{R}^k)$ w nią samą. Ponadto

$$\nabla_\sigma \circ \nabla_{-\sigma} = \mathrm{id}^{8}, \qquad \nabla^*_\sigma \circ \nabla^*_\sigma = \mathrm{id}, \qquad \nabla^*_\sigma = \nabla^* \circ \nabla_\sigma = \nabla_{-\sigma} \circ \nabla^* \quad \text{oraz} \quad \nabla^*_0 = \nabla^*.$$

Zauważmy również, że przesunięcie funkcji $\nabla_\sigma(u)$ jest funkcją u przesuniętą w lewo o σ, obrót funkcji $\nabla^*(u)$ jest funkcją u obróconą o π wokół osi rzędnych, a splotowe przesunięcie funkcyjne ∇^*_σ jest złożeniem operatorów ∇^* z ∇_σ, przy czym złożenia tego nie można wykonać w dowolnej kolejności (G. Ciesielski [14], s. 17, oraz [11] i [12]).

Przejdźmy teraz do definicji podstawowych właściwości operatorów opisujących systemy.

Operator ϕ na $\mathbf{U}_F \subset \mathbf{L}^m{}_F(\mathbf{R}^k)$ nazywamy stacjonarnym, jeżeli

$$\forall u \in \mathbf{U} \forall \sigma \in \mathbf{R}\big((\nabla_\sigma(u) \in \mathbf{U}) \wedge (\nabla_\sigma(\phi(u)) = \phi(\nabla_\sigma(u)))\big).$$

Innymi słowy, operator jest stacjonarny, gdy odpowiedź operatora na wymuszenie u przesunięte w czasie o σ jest równa przesuniętej w czasie o σ odpowiedzi operatora

[7] Przez := oznaczamy równość definicyjną.

[8] Przez id oznaczamy odwzorowanie identycznościowe (identyczność).

na wymuszenie u. Zauważmy, że przestrzeń liniowa U musi być zbiorem zamkniętym ze względu na przsunięcia funkcyjne, a zatem zawierać musi wszystkie przesunięcia funkcyjne swoich elementów. Przestrzeń o takiej właściwości będziemy nazywać *przestrzenią stacjonarną*. Kolejną ważną właściwością operatora jest przyczynowość.

Operator ϕ na $\mathbf{U}_F \subset \mathbf{L}^m{}_F(\mathbf{R}^k)$ nazywamy *przyczynowym*, jeżeli

$$\forall u,\ v \mathrm{bU}\ \forall t \in \mathbf{R}\big(\forall \tau \in \mathbf{R}\big((\tau \leqslant t) \Rightarrow (u(\tau) = v(\tau))\big) \Rightarrow \big([\phi(u)](t) = [\phi(v)](t)\big)\big).$$

Wynika stąd, że operator jest przyczynowy, gdy wartość sygnału wyjściowego w chwili t zależy jedynie od wartości sygnału wejściowego do chwili t.

Wprowadźmy teraz definicje pewnych przestrzeni, które będziemy wykorzystywać dalej. Przestrzenie te będą stanowiły podstawę opisu przestrzeni wymuszeń i odpowiedzi rozważanych systemów.

Załóżmy, że μ jest miarą na σ-ciele w X.

Jeżeli $1 \leqslant p < \infty$, to *przestrzenią* $\mathbf{L}^m{}_{Fp\mu}(X)$ nazywamy zbiór klas równoważności ze względu na relację $\mu - \forall\exists$ [9], funkcji mierzalnych (μ-mierzalnych) $u{:}X \mapsto \mathbf{F}^m$ takich, że

$$\int_X |u|^p d\mu < \infty .$$

Normą w $\mathbf{L}^m{}_{Fp\mu}(X)$ nazywamy funkcję, którą $\forall u \in \mathbf{L}^m{}_{Fp\mu}(X)$ oznaczamy przez

$$\|u\|_p := \left(\int_X |u|^p \, d\mu\right)^{1/p}.$$

Funkcje należące do $\mathbf{L}^m{}_{F1\mu}(X)$ nazywamy *całkowalnymi* (w sensie Labesgue'a).

Przestrzenią $\mathbf{L}^m{}_{F\infty\mu}(X)$ nazywamy zbiór klas równoważności ze względu na relację $\mu - \forall\exists$, funkcji mierzalnych (μ-mierzalnych) $u{:}X \mapsto \mathbf{F}^m$, dla których

$$\operatorname*{supess}_{x \in X} |u(x)| < \infty ,$$

gdzie

$$\operatorname*{supess}_{x \in X} |u(x)| := \inf\{M \in \overline{\mathbf{R}}_+ : \mu - \forall\exists x \in X(|u(x)| \leqslant M\}$$ [10],

nazywamy *istotnym kresem górnym* funkcji $|u|$. *Normą* w $\mathbf{L}^m{}_{p\infty\mu}(X)$ nazywamy funkcję, którą $\forall u \in \mathbf{L}^m{}_{F\infty\mu}(X)$ oznaczamy przez

$$\|u\|_\infty := \operatorname*{supess}_{x \in X} |u(x)| .$$

Funkcje należące do przestrzeni $\mathbf{L}^m{}_{F\infty\mu}(X)$ nazywamy *istotnie* (μ-*istotnie*) *ograniczonymi* (J. Musielak [36], s. 53 i 59).

[9] Przez $\mu - \forall\exists$ oznaczamy zwrot „μ-prawie wszędzie" lub „dla μ-prawie każdego".

[10] Przez $\overline{R}_+$ oznaczamy zbiór liczb rzeczywistych nieujemnych uzupełniony o liczbę ∞.

Przestrzenią $\mathbf{L}^m{}_{F2\mu}(X)$ nazywamy zbiór klas równoważności ze względu na relację $\mu - \forall\exists$, funkcji mierzalnych (μ-mierzalnych) $u{:}X \mapsto \mathbf{F}^m$, dla której *iloczynem skalarnym* nazywamy funkcję, którą $\forall u, v \in \mathbf{L}^m_{F2\mu}(X)$ oznaczamy przez

$$\langle u, v\rangle := \int_X v^* u \mathrm{d}\mu \,[11].$$

Normą wyznaczaną przez iloczyn skalarny w $\mathbf{L}^m{}_{F2\mu}(X)$ nazywamy funkcję, którą $\forall u \in \mathbf{L}^m_{F2\mu}(X)$ oznaczamy przez

$$\| u \|_2 := (\langle u, u\rangle)^{1/2}.$$

Dla dowolnego układu $u_{[k]} := (u_i \in \mathbf{L}^m{}_{F2\mu}(X))_{i=1}^k$, *macierzy Grama* oznaczną przez

$$G(u_{[k]}) := (\langle u_i, u_j\rangle)_{i,j=1}^k.$$

Załóżmy dalej, że $1 \leqslant p \leqslant \infty$.

Przez $\mathbf{C}^m_{Fp\mu}(X)$ będziemy oznaczać podprzestrzeń funkcji ciągłych z przestrzeni $\mathbf{L}^m{}_{Fp\mu}(X)$.

Jeżeli zbiór X wynika z kontekstu, to będziemy pisać w skrócie $\mathbf{L}^m_{Fp\mu}$ lub $\mathbf{C}^m_{Fp\mu}$. Oznaczenia $\mathbf{L}^1{}_{Fp\mu}$ i $\mathbf{C}^1_{Fp\mu}$ oraz odpowiednio $\mathbf{L}_{Fp\mu}$ i $\mathbf{C}_{Fp\mu}$ będziemy traktować jako równoważne. W przypadku gdy miara μ jest miarą Lebesgue'a na $\mathbf{R}^k$, którą będziemy oznaczać przez λ_k, w skrócie λ, to oznaczenia $\mathbf{L}^m{}_{Fp\lambda}$ i $\mathbf{C}^m_{Fp\lambda}$ oraz odpowiednio $\mathbf{L}^m{}_{Fp}$ i $\mathbf{C}^m_{Fp}$, jak też $\lambda - \forall\exists$ oraz $\forall\exists$, są równoważne.

Przejdźmy teraz do definicji uogólnionego splotu funkcji.

Niech λ_k będzie miarą Lebesgue'a na $\mathbf{R}^k$. *Splotem funkcji* mierzalnych u, $v{:}\mathbf{R}^k \mapsto \mathbf{F}^m$ nazywać będziemy funkcję oznaczoną przez $u^* v{:}\mathbf{R}^k \mapsto \mathbf{F}$, która o ile istnieje spełnia warunek:

$$\lambda_k - \forall\exists t \in \mathbf{R}^k \left([u^* v](t) := \int_{\mathbf{R}^k} u^T(t-\tau) v(\tau) d\tau \right).$$

Zwykle splot definiuje się dla funkcji o wartściach z $\mathbf{F}$, a nie tutaj z $\mathbf{F}^m$. Takie rozszerzenie pojęcia splotu wynika z jego związku z iloczynem skalarnym (G. Ciesielski [14]), s. 33 oraz [13] i [17]). Zauważmy również, że całka Lebesgue'a może przyjmować watości nieskończone, które nie należą do ciała $\mathbf{F}$ [12], podczas gdy wartości splotu funkcji należą do $\mathbf{F}$. Z warunku definicji wynika, że splot funkcji itnieje, gdy zbiór tych $t \in \mathbf{R}^k$, dla których całka przyjmuje wartości spoza $\mathbf{F}$ jest miary zero. Główne właściwości tak zdefiniowanych splotów omówiono w [14], s. 29, oraz [13] i [17].

[11] Dla dowolnej macierzy A przez A^* oznaczamy jej transpozycję ze sprzężeniem, a więc

$$A^* := \overline{A^T}.$$

[12] Załóżmy dla uproszczenia, że $\mathbf{F} = \mathbf{R}$. Mówimy, że *istnieje całka Lebesgue'a* funkcji μ-mierzalnej $f{:}X \mapsto \mathbf{R}$, jeżeli

$$\int_X f d\mu \in \overline{\mathbf{R}},$$

gdzie przez $\overline{\mathbf{R}}$ oznaczamy zbiór liczb rzeczywistych R uzupełniony o liczby $-\infty$ i $+\infty$. Z istnienia całki Lebesgue'a nie wynika całkowalność funkcji f.

Przekonamy się dalej, że przestrzenie stacjonarne, splotowe i przyczynowe odgrywają bardzo ważną rolę w modelowaniu systemów stacjonarnych.

Załóżmy, że μ jest miarą na $\mathbf{R}^k$ oraz $1 \leqslant p \leqslant \infty$.

Przestrzenią stacjonarną oznaczaną przez $\mathbf{L}^m_{Fp\mu\nabla}(\mathbf{R}^k)$, nazywamy podzbiór klas równoważności z przestrzeni $\mathbf{L}^m{}_{Fp\mu}(\mathbf{R}^k)$ ze względu na relację $\mu - \forall\exists$ o postaci:

$$\mathbf{L}^m{}_{Fp\mu\nabla}(\mathbf{R}^k) := \{u \in \mathbf{L}^m{}_{Fp\mu}(\mathbf{R}^k) : \forall t \in \mathbf{R}^k (\nabla_t(u) \in \mathbf{L}^m{}_{Fp\mu}(\mathbf{R}^k))\}.$$

Przestrzenią splotową oznaczaną przez $\mathbf{L}^m{}_{Fp\mu^*}(\mathbf{R}^k)$, nazywamy podzbiór klas równoważności z przestrzeni $\mathbf{L}^m{}_{Fp\mu}(\mathbf{R}^k)$, ze względu na relację $\mu - \forall\exists$ o postaci:

$$\mathbf{L}^m{}_{Fp\mu^*}(\mathbf{R}^k) := \{u \in \mathbf{L}^m{}_{Fp\mu}(\mathbf{R}^k) : \forall t \in \mathbf{R}^k (\nabla^*_t(u) \in \mathbf{L}^m{}_{Fp\mu}(\mathbf{R}^k))\}.$$

Przestrzenią przyczynową oznaczaną przez $\mathbf{L}^m_{Fp\mu+}(\mathbf{R}^k)$, nazywamy podzbiór klas równoważności z przestrzeni $\mathbf{L}^m{}_{Fp\mu}(\mathbf{R}^k)$ ze względu na relację $\mu - \forall\exists$ o postaci:

$$\mathbf{L}^m{}_{Fp\mu+}(\mathbf{R}^k) := \{u_{-1}u : u \in \mathbf{L}^m{}_{Fp\mu}(\mathbf{R}^k)\}^{13}.$$

Właściwości tych przestrzeni są omówione w [14], s. 32, oraz [13] i [17].

Przez $\mathbf{C}^m_{Fp\mu\nabla}(X)$, $\mathbf{C}^m_{Fp\mu^*}(X)$ i $\mathbf{C}^m_{Fp\mu+}(X)$ będziemy oznaczać podprzestrzeń funkcji ciągłych odpowiednio z przestrzeni $\mathbf{L}^m{}_{Fp\mu\nabla}(X)$, $\mathbf{L}^m{}_{Fp\mu^*}(X)$ i $\mathbf{L}^m{}_{Fp\mu+}(X)$.

Jeżeli zbiór $\mathbf{R}^k$ wynika z kontekstu, to będziemy pisać w skrócie $\mathbf{L}^m{}_{Fp\mu\nabla}$, $\mathbf{L}^m{}_{Fp\mu^*}$, $\mathbf{L}^m{}_{Fp\mu+}$, $\mathbf{C}^m_{Fp\mu\nabla}$, $\mathbf{C}^m_{Fp\mu^*}$ oraz $\mathbf{C}^m_{Fp\mu+}$. Oznaczenia $\mathbf{L}^1{}_{Fp\mu\nabla}$, $\mathbf{L}^1{}_{Fp\mu^*}$, $\mathbf{L}^1{}_{Fp\mu+}$, $\mathbf{C}^1_{Fp\mu\nabla}$, $\mathbf{C}^1_{Fp\mu^*}$ i $\mathbf{C}^1_{Fp\mu+}$ oraz odpowiednio $\mathbf{L}_{Fp\mu\nabla}$, $\mathbf{L}_{Fp\mu^*}$, $\mathbf{L}_{Fp\mu+}$, $\mathbf{C}_{Fp\mu\nabla}$, $\mathbf{C}_{Fp\mu^*}$ i $\mathbf{C}_{Fp\mu+}$ będziemy traktować jako równoważne. W przypadku gdy miara na $\mathbf{R}^k$ jest miarą Lebesgue'a, to jej symbol będzie pomijany w oznaczeniach zdefiniowanych przestrzeni.

Zauważmy, że zgodnie z definicją stacjonarności operatora, sformułowaną w tej części artykułu, przestrzeń wymuszeń operatora stacjonarnego musi zawierać wszystkie przesunięcia funkcyjne swoich elementów. Przestrzeń stacjonarna $\mathbf{L}^m{}_{Fp\mu\nabla}(\mathbf{R})$ ma tą właściwość, co pozwala nam definiować na niej operatory stacjonarne. Zauważmy tu również, że podobnie przestrzeń splotowa $\mathbf{L}^m{}_{Fp\mu^*}(\mathbf{R})$ zawiera wszystkie splotowe przesunięcia funkcyjne swoich elementów. Z kolei elementy składające się na przestrzeń przyczynową $\mathbf{L}^m{}_{Fp\mu+}(\mathbf{R})$ są funkcjami, które przyjmują wartość 0 dla ujemnych wartości zmiennej niezależnej. Sygnały należące do tej przestrzeni nazywane są przyczynowymi (A. Wojnar [55], s. 48).

[13] Przez u_{-1} oznaczamy *skok jednostkowy*, który $\forall t := (t_i \in \mathbf{R})_1^k$ zdefiniowany jest przez wzór:

$$u_{-1} := \begin{cases} 1, & \forall i \in \{1,\ldots,k\}(t_i \geqslant 0) \\ 0, & \exists i \in \{1,\ldots,k\}(t_i < 0) \end{cases}.$$

Jeżeli $u : X \mapsto \mathbf{F}$ oraz $v : X \mapsto \mathbf{F}^n$, to $\forall x \in X$ *iloczyn funkcji* u i v oznaczany przez uv, zdefiniowany jest za pomocą wzoru:

$$[uv](x) := u(x)v(x).$$

Można wykazać (G. Ciesielski [14], s. 18 oraz [11] i [12]), że dowolny operator ϕ na $\mathbf{L}^m{}_{F2\kappa\mu\nabla}(\mathbf{R})$ jest stacjonarny, wtedy i tylko wtedy, gdy

$$\phi = \varphi \circ \nabla ,$$

gdzie przekształcenie

$$\varphi := [\phi(\cdot)](0) .$$

Z kolei $\forall t \in \mathbf{R}$ obrazem przestrzeni $\mathbf{L}^m{}_{F2\kappa\nabla}(\mathbf{R})$ w odwzorowaniu ∇_t jest ta sama przestrzeń $\mathbf{L}^m{}_{F2\kappa\nabla}(\mathbf{R})$, a zatem obrazem produktu $\mathbf{L}^m{}_{F2\kappa\nabla}(\mathbf{R}) \times \mathbf{R}$ w odwzorowaniu ∇. jest również przestrzeń $\mathbf{L}^m{}_{F2\kappa\nabla}(\mathbf{R})$. Jest to bardzo ważny wynik, który pozwala ograniczyć analizę struktury dowolnego stacjonarnego operatora ϕ na $\mathbf{L}^m{}_{F2\kappa\nabla}(\mathbf{R})$, do analizy struktury odpowiadającego mu przekształcenia φ również na $\mathbf{L}^m{}_{F2\kappa\nabla}(\mathbf{R})$ (G. Ciesielski [14], s. 18).

Załóżmy, że $$\mathbf{L}^m{}_{F2\kappa\nabla}(\mathbf{R}) = \mathbf{L}^m{}_{F2\kappa^*}(\mathbf{R}) = \mathbf{L}^m_{F2\kappa}(\mathbf{R})$$

oraz $\varphi \in \mathbf{C}^n_{F\infty\mu}(\mathbf{L}^n_{F2\kappa}(\mathbf{R}))$ jest przekształceniem ograniczonym, dla którego złożenie $\varphi \circ \nabla$. jest operatorem stcjonarnym przyczynowym. Wynika wówczas, że każdy operator $\varphi \circ \nabla$. można dowolnie dokładnie przybliżyć za pomocą liniowej kombinacji operatorów o postaci:

$$[\vartheta_{k[l]}(u)](t) := \left([\vartheta_{k,j}(u)](t)\right)_1^l := \vartheta_{k[l]}(\nabla_t(u)) = \chi_{[l]}\left([\overline{wv}_{[k]} * u](t)\right),$$

gdzie $\chi_{[l]}$ jest układem liniowo niezależnym w $\mathbf{L}^n{}_{F2v_k}(\mathbf{F}^k)$, a $v_{[k]}$ jest układem ortonormalnym w $\mathbf{L}^m{}_{F2\kappa+}(\mathbf{R})$ (G. Ciesielski [14], s. 51, oraz [16] i [20]).

3. ZADANIE MODELOWANIA

Zadanie modelowania dowolnego systemu ϕ ma na celu wyznaczenie matematycznego modelu rozważanego systemu. Zadanie to sklada się z *identyfikacji strukturalnej* oraz *identyfikacji parametrycznej* modelu systemu (J. Jaworski, R. Morawski, J. Olędzki [31]). Zadanie identyfikacji strukturalnej daje w wyniku strukturę poszukiwanego systemu, a zadanie identyfikacji parametrycznej pokazuje jak wyznaczać parametry struktury będącej wynikiem identyfikacji strukturalnej. Omówimy teraz metody wyznaczania parametrów struktury będącej wynikiem identyfikacji strukturalenj oraz identyfikacji parametrycznej. Otrzymane wyniki wykorzystamy do modelowania systemu ϕ.

4. IDENTYFIKACJA STRUKTURALNA

Sformułujemy teraz twierdzenie rozwiązujące problem identyfikacji strukturalnej pewnej bardzo ważnej klasy liniowych operatorów stacjonarnych przyczynowych.

Twierdzenie 4.1. *Dla dowolnego liniowego ciągłego operatora stacjonarnego przyczynowego* $\phi: \mathbf{C}_{\mathbf{R}\infty+}(\mathbf{R})$, *prawdziwe są następujące twierdzenia.*

(1) *Istnieje funkcja* $g \in \mathbf{L}_{R1+}(\mathbf{R})$ *taka, że* $\forall u \in \mathbf{C}_{R\infty+}(\mathbf{R})$ *i* $\forall t \in \mathbf{R}$

$$[\phi(u)](t) = [g^* u](t) := \int_R g(t-\tau)u(\tau)\mathrm{d}\tau .$$

Załóżmy dalej, że $g \in \mathbf{L}_{R1+}(\mathbf{R})$ *jest funkcją, o której mowa w części* (1).
(2) *Jeżeli funkcja g jest istotnie ograniczona, czyli* $g \in \mathbf{L}_{R\infty+}(\mathbf{R})$, *to* $g \in \mathbf{L}_{R2+}(\mathbf{R})$.
(3) *Jeżeli* $\langle \cdot, \cdot \rangle$ *jest iloczynem skalarnym w* $\mathbf{L}_{R2}(\mathbf{R})$, $v_{[k]} := (v_i)_1^k$ *jest układem liniowo niezależnym w* $\mathbf{L}_{R2+}(\mathbf{R})$ *oraz* $g \in \mathbf{L}_{R2+}(\mathbf{R})$, *to*

$$g_k := (G^{-1}(v_{[k]})\langle g, v_{[k]}\rangle)^T v_{[k]} := (G^{-1}(v_{[k]})(\langle g, v_i\rangle)_1^k)^T v_{[k]},$$

jest przybliżeniem średniokwadratowym funkcji g, w którym

$$G(v_{[k]}) := (\langle v_i, v_j\rangle)_{i,j=1}^k \text{ [14]}.$$

Dowód.
(1) Dowód tej części wynika z twierdzenia o modelowaniu, z jednolitej ogólnej teorii operatorowej systemów (G. Ciesielski [14], s. 51, [16], i [20]). Można go również znaleźć w książce J. Kudrewicza [33], s. 76, przy dodatkowym założeniu, że system jest ograniczony lub ciągły.
(2) Niech $\|\cdot\|_1$, $\|\cdot\|_2$ oraz $\|\cdot\|_\infty$ będą normami odpowiednio w $\mathbf{L}_{R1}$, $\mathbf{L}_{R2}$ oraz $\mathbf{L}_{R\infty}$. Ponieważ z założenia $g \in \mathbf{L}_{R1+} \cap \mathbf{L}_{R\infty+}$, to

$$\|g\|_2 = \int_R |g|^2 = \int_R |g|\ |g| \leqslant \|g\|_\infty \int_R |g| = \|g\|_\infty \|g\|_1 = \|g\|_1\ \|g\|_\infty < \infty ,$$

a zatem $g \in \mathbf{L}_{R2}$. Stąd i z części (1) wynika ostatecznie, że $g \in \mathbf{L}_{R2+}$.
(3) Dowód tej części w formie uproszczoenj, można znaleźć w książce G. Dahlquista, Å. Björcka [25], s. 91. ■

Z udowodnionego twierdzenia wynika, że każdy liniowy ciągły system stacjonarny przyczynowy $\phi: \mathbf{C}_{R\infty+} \mapsto \mathbf{C}_{R\infty+}$ można opisać za pomocą splotu funkcji, a zatem

$$\phi(u) = g^* u ,$$

gdzie g jest odpowiedzią impulsową tego systemu. Odpowiedź impulsowa g jest całkowalna, a gdy ponadto odpowiedź g jest istotnie ograniczona, to jest ona również całkowalna w kwadracie. Jest to bardzo ważne twierdzenie należące w analizie funkcjonalnej do grupy twierdzeń o reprezentacjach odwzorowań. Twierdzenia należące do tej grupy dotyczą zadania identyfikacji strukturalnej różnego rodzaju odwzorowań.

Zauważmy, że założenie w twierdzeniu 4.1 ciągłości $\mathbf{R}$ odpowiedzi systemu ϕ nie jest zbyt drastyczne, gdy każdy fizycznie realizowany system ma skończone pasmo przenoszenia, a stąd wynika ciągłość na $\mathbf{R}$ jego odpowiedzi. Podobnie założenie ograniczoności na $\mathbf{R}$ odpowiedzi systemu ϕ wynika ze skończonej mocy układów

[14] Macierz ta jest nazywana macierzą Grama.

zasilających ten system. Zauważmy tu również, że zgodnie z definicją stacjonarności operatora, sformułowaną w części 2, przestrzeń wymuszeń operatora stacjonarnego musi zawierać wszystkie, w tym przypadku ujemne ($\sigma \leqslant 0$), przesunięcia funkcyjne swoich elementów. Przestrzeń $\mathbf{C}_{\mathbf{R}\infty+}$ ma tą właściwość, co pozwala nam definiować na niej operatory stacjonarne.

Udowodnione twierdzenie jest szczególnym przypadkiem twierdzenia o modelowaniu układów liniowych (G. Ciesielski [13] i [17]). Jest ono na tyle proste, iż pozwala ominąć na tym etapie wszystkie złożoności związane z teorią miery i całki Lebesgue'a.

5. IDENTYFIKACJA PARAMETRYCZNA

Zajmiemy się teraz identyfikacją parametryczną liniowych ciągłych systemów stajonarnych $\phi: \mathbf{C}_{\mathbf{R}\infty+} \mapsto \mathbf{C}_{\mathbf{R}\infty+}$ o całkowalnych w kwadracie odpowiedziach impulsowych. Z twierdzenia 4.1 (3) wynika, że każdy taki system ϕ można przybliżyć za pomocą kombinacji liniowych systemów stacjonarnych przyczynowych $\phi_{[k]} := (\phi_i)_1^k$, których odpowiedzi impulsowe $(v_i)_1^k =: v_{[k]}$ tworzą układ liniowo niezależny w $\mathbf{L}_{\mathbf{R}2+}$. Zauważmy bowiem, że $\forall u \in \mathbf{C}_{\mathbf{R}\infty+}$

$$\phi_k(u) := g_k * u = (c_{[k]}^T v_{[k]}) * u = c_{[k]}^T (v_{[k]} * u) := c_{[k]}^T v_i * u)_1^k,$$

gdzie

$$c_{[k]} := (c_i \in \mathbf{R})_1^k = G^{-1}(v_{[k]}) \langle g, v_{[k]} \rangle := G^{-1}(v_{[k]})(\langle g, v_i \rangle)_1^k.$$

Przybliżenie ϕ_k można więc schematycznie przedstawić tak, jak to zostało pokazane na rys. 5.

1. Identyfikacja parametryczna sprowadza się więc do postępowania mającego na celu wyznaczenie współczynników kombinacji liniowej, które składają się na wektor $c_{[k]}$. Wymaga to jak widać wyznaczenia odpowiednich iloczynów skalarnych w $\mathbf{L}_{\mathbf{R}2}$.

Jeżeli system ϕ jest dany w postaci analitycznej, to zadanie identyfikacji parametrycznej jest dość prostym problemem obliczeniowym, polegającym na rozwiązaniu typowego zadania aproksymacji średniokwadratowej. Zajmiemy się zatem dalej systemami danymi w postaci nieanalitycznej, a ściślej, jedną z najważniejszych reprezentacji tej postaci, dla której mamy dostępne jedynie wejście i wyjście systemu ϕ.

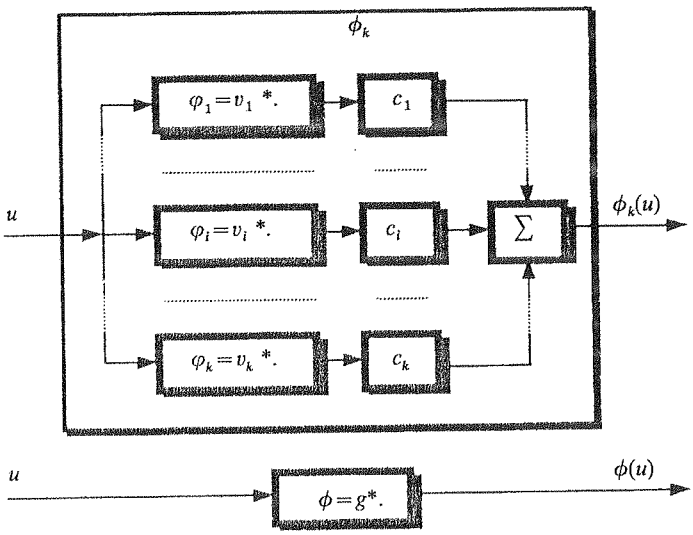
Kolejne twierdzenie pozwala rozwiązać zadanie identyfikacji parametrycznej.

Twierdzenie 5.1. *Jeżeli* $\langle \cdot, \cdot \rangle$ *jest iloczynem skalarnym w* $\mathbf{L}_{\mathbf{R}2}(\mathbf{R})$ *oraz* $x: \mathbf{R} \mapsto \mathbf{R}$ *jest ergodycznym szumem białym o gęstości widmowej mocy* S_x, *to* $\forall u, v \in \mathbf{L}_{\mathbf{R}2}(\mathbf{R})$

$$A((u*x)(v*x)) = S_x \langle u, v \rangle \,^{15}.$$

[15] Wartość średnia sygnału mierzalnego $u: \mathbf{R} \mapsto \mathbf{R}$, oznaczana dalej przez

$$A(u) := \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} u(t) dt.$$



Rys. 5. 1. Struktura przybliżenia ϕ_k liniowego ciągłego systemu stacjonarnego przyczynowego ϕ: $C_{R\infty+} \mapsto C_{R\infty+}$ o całkowalnej w kwadracie odpowiedzi impulsowej.

Dowód. Przypomnijmy, że ergodyczny szum biały ma stałą gęstość widmową mocy $\sqrt{S_x} = \text{const}$, gdyż wówczas $\forall t \in \mathbf{R}$ autokorelacja szumu x ma postać:

$$R_x(t) := A(\nabla_t(x)x) := \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t+\tau)x(\tau)d\tau = \mathscr{F}^{-1}(S_x) = S_x u_0(t)\,^{16}.$$

Wartość średnia

$$A((u*x)(v*x)) := \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} [u*x](t)[v*x](t)dt = \lim_{T-\infty} \frac{1}{2T} \int_{-T}^{T} [x*u](t)[x*v](t)dt =$$

$$= \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} \left(\int_{\mathbf{R}} x(t-\tau)u(\tau)d\tau \int_{\mathbf{R}} x(t-\tau)v(\tau)d\tau \right) dt =$$

$$= \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} \left(\iint_{\mathbf{RR}} (x(t-\tau_1)u(\tau_1))(x(t-\tau_2)v(\tau_2))d\tau_1 d\tau_2 \right) dt =$$

$$= \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} \left(\iint_{\mathbf{RR}} (u(\tau_1)x(t-\tau_1))(x(t-\tau_2)v(\tau_2))d\tau_1 d\tau_2 \right) dt =$$

$$= \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} \left(\iint_{\mathbf{RR}} u(\tau_1)(x(t-\tau_1)x(t-\tau_2))v(\tau_2)d\tau_1 d\tau_2 \right) dt =$$

[16] Przez $\mathscr{F}$ oznaczamy przekształcenie Fouriera, a przez u_0, impuls Diraca.

$$= \iint_{RR} u(\tau_1)\left(\lim_{T\to\infty}\frac{1}{2T}\int_{-T}^{T} x(t-\tau_1)x(\tau_2)dt\right)v(\tau_2)d\tau_1 d\tau_2 =$$

$$= \iint_{RR} u(\tau_1)((S_x u_0(\tau_2-\tau_1))v(\tau_2)d\tau_1 d\tau_2 =$$

$$= S_x \int_R \left(\int_R u(\tau_1)u_0(\tau_2-\tau_1)d\tau_1\right)v(\tau_2)d\tau_2 = S_x \int_R u(\tau_2)v(\tau_2)d\tau_2 = S_x\langle u, v\rangle ,$$

a stąd otrzymujemy tezę. ■

Z udowodnionego twierdzenia można wywnioskować, że wektor współczynników kombinacji liniowej

$$c_{[k]} := (c_i)_1^k = G^{-1}(v_{[k]})\langle g, v_{[k]}\rangle := G^{-1}(v_{[k]})(\langle g, v_i\rangle)_1^k =$$

$$\left(\frac{1}{S_x}(A((v_i{}^*x)(v_j{}^*x)))_{i,j=1}^k\right)^{-1}\frac{1}{S_x}(A((g^*x)(v_i{}^*x)))_1^k =$$

przyczynowego

$$= ((A((v_i{}^*x)(v_j{}^*x)))_{i,j=1}^k)^{-1}(A((g^*x)(v_i{}^*x)))_1^k = (A((v_{[k]}{}^*x)(v_{[k]}{}^*x)^T))^{-1}A((g^*x)(v_{[k]}{}^*x)) .$$

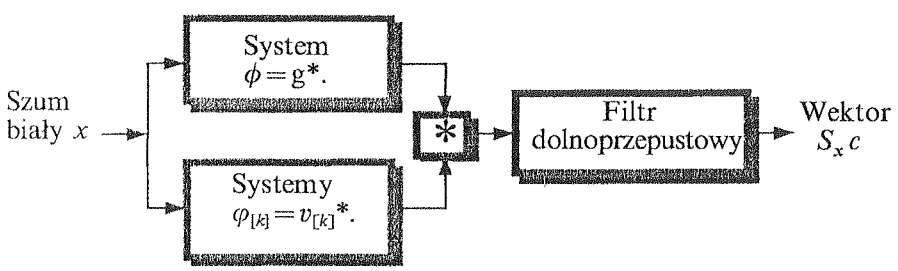
ść widmową
stać:

Jeżeli założymy, że układ $v_{[k]}$ jest ortonormalny, to macierz Grama $G(v_{[k]})$ jest macierzą jedynkową i wektor

$$c_{[k]} := (c_i)_1^k = \langle g, v_{[k]}\rangle := (\langle g, v_i\rangle)_1^k = \frac{1}{S_x}(A((g^*x)(v_i{}^*x)))_1^k =: \frac{1}{S_x}A((g^*x)(v_{[k]}{}^*x)) .$$

)[16].

(t)dt=



Rys. 5. 2. Schemat blokowy do eksperymentalnego wyznaczania wektora c.

Wynika stąd, że wektor ten może być wyznaczony w bardzo prostym układzie, którego schemat blokowy przedstawiono na rys. 5.2. Schemat ten również odczytać jako schemat blokowy algorytmu wyznaczania wektora c.

6. BAZA ORTONORMALNA W PRZESTRZENI L_{R2+}

Pojawiający się tu problem polega na wyznaczaniu postaci układu liniowo niezależnego $v_{[k]} := (v_i)_1^k$ w L_{R2+}, o którym była mowa w twierdzeniu 4.1. (3).

Wielomianem Laguerre'a stopnia n (n-tym) oznaczanym przez $L_{c,n}$, nazywamy wielomian stopnia n, który $\forall t \in \mathbf{R}$ określony jest przez wzory:

$$L_{c,0}(t) := \sqrt{2c}, \qquad L_{c,1}(t) := \sqrt{2c}(2ct-1) \text{ oraz}$$

$$nL_{c,n}(t) := (2ct-2n+1)L_{c,n-1}(t) - (n-1)L_{c,n-2}(t),$$

gdzie $c \in]0,\infty[$ jest stałą nazwą *współczynnikiem skali (czasu)*. Z kolei *funkcją Laguerre'a (n-tą)* oznaczaną przez $l_{c,n}$, nazywamy funkcję, która $\forall t \in \mathbf{R}$ określona jest przez wzór:

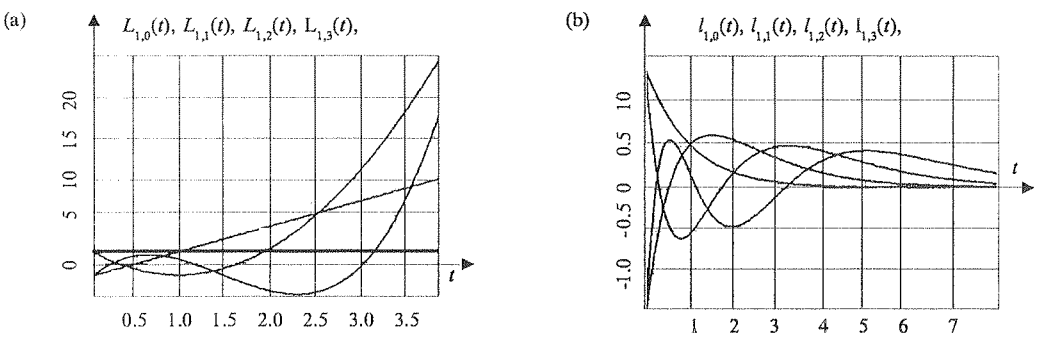
$$l_{c,n}(t) := L_{c,n}(t)\exp(-ct)u_{-1}(t),$$

gdzie funkcja u_{-1} skokiem jednostkowym. Jeżeli współczynnik c wynikać będzie z kontekstu, to dopuszczamy również skrócone oznaczenia L_n oraz l_n.

Twierdzenie 6.1. *Układ funkcji Laguerre'a* $(l_{c,i})_N$ *jest bazą ortonormalną w przestrzeni* $\mathbf{L}_{R2+}$.

Dowód podano za J. Szabatinem [53], s. 125.■

Wynika stąd, że funkcje te mogą być użyte do budowy układu liniowo niezależnego $v_{[k]} := (v_i)_1^k$, o którym była mowa w twierdzeniu 4.1. (3). Zauważmy bowiem, że każdy układ ortogonalny, a zatem i ortonormalny, jest liniowo niezależny (J. Musielak [36], s. 70). Definicja funkcji Laguerre'a wykorzystuje pojęcie wielomianów Laguerre'a.



Rys. 6. 1. Wykresy pierwszych czterech wielomianów (a) i funkcji Laguerre'a (b).

Wykresy pierwszych czterech wielomianów i funkcji Laguerre'a pokazano na rys. 6.1, a wynikającą teraz strukturę przybliżenia ϕ_k liniowego ciągłego systemu stacjonarnego przyczynowego $\phi: \mathbf{C}_{R\infty+} \mapsto \mathbf{C}_{R\infty+}$ o całkowalnej w kwadracie odpowiedzi impulsowej, pokazano na rys. 6.2. Przybliżenie ϕ_k o tej postaci nazywane jest *modelem Laguerre'a*. Zauważmy, że w modelu tym układ liniowo niezależny

$$v_{[k]} := (v_i)_1^k = (l_{c,i-1})_1^k,$$

a zatem model Laguerre'a jest kombinacją liniową systemów stacjonarnych przyczynowych $\lambda_{[k]} := (\lambda_{c,i})_1^k$ nazywanych *systemami Laguerre'a*, których odpowiedzi impulsowe $(l_{c,i-1})_1^k =: v_{[k]}$ tworzą układ liniowo niezależny w $\mathbf{L}_{R2+}$.

Rys. 6.2. M

Warto
transform
Laguerre'

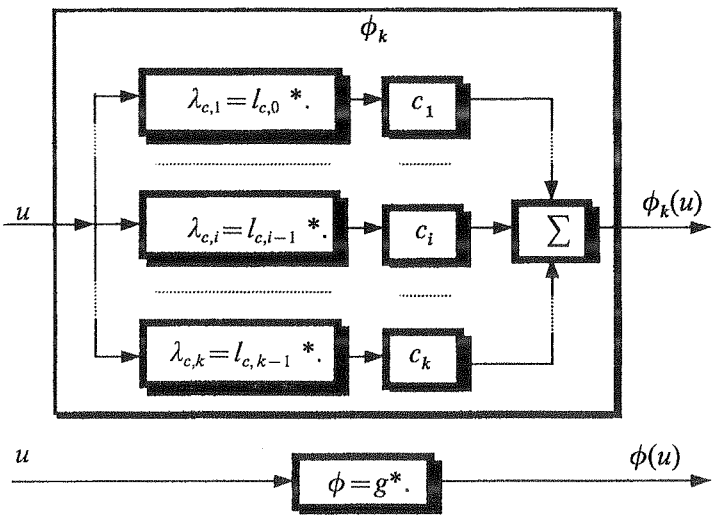
(M. Schet
(J. Osiow
Laguerre'

Z liniowo
liwią num
do przest
Amierbaj

7. P

Jakoś
metrów (

[17] Przez



Rys. 6.2. Model Laguerre'a ϕ_k liniowego systemu stacjonarnego przyczynowego ϕ: $C_{R\infty+} \mapsto C_{R\infty+}$ o całkowalnej w kwadracie odpowiedzi impulsowej.

Warto w tym miejscu wspomnieć, że wielomiany oraz funkcje Laguerre'a mają transformatory Laplace'a. Wynika to z faktu, że transformata Laplace'a funkcji Laguerre'a oznaczana przez

$$[\mathscr{L}(l_{c,n})](s)=\frac{\sqrt{2c}}{c+s}\left(\frac{c-s}{c+s}\right)^n \text{ [17]}$$

(M. Schetzen [48], s. 352), a stąd i z twierdzenia o przesunięciu w dziedzinie zespolonej (J. Osiowski [37], s. 138) mamy, że jednostronna transformata Laplace'a wielomianu Laguerre'a

$$[\mathscr{L}(L_{c,n})](s)=[\mathscr{L}(l_{c,n}\exp(c\cdot))](s)=\frac{\sqrt{2c}}{s}\left(\frac{2c-s}{s}\right)^n.$$

Z liniowości przekształcenia Laplace'a wynika zatem, że funkcja Laguerre'a umożliwią numeryczne wyznaczanie transformat Laplace'a dowolnych funkcji należących do przestrzeni L_{R2+}, gdyż stanowią dla niej bazę ortonormalną (patrz też W. M. Amierbajew, N. A. Utiembajew [1], s. 9).

7. PROGRAM TURBO WIENER™ DO WYZNACZANIA MODELI LINIOWYCH SYSTEMÓW STACJONARNYCH

Jakość oraz efektywność przedstawionych wcześniej metod wyznaczania parametrów (identyfikacji parametrycznej) modeli liniowych ciągłych systemów stac-

[17] Przez $\mathscr{L}$ oznaczamy przekształcenie Laplace'a.

jonarnych przyczynowych o całkowalnej w kwadracie odpowiedzi impulsowej, można przebadać za pomocą programu Turbo Wiener™.

Jednym z problemów jakie należałoby rozwiązać w trakcie pisania programu Turbo Wiener™, był sposób opisu systemu modelowanego. W programie Turbo Wiener™ założono, że system ten jest opisany za pomocą liniowych równań różniczkowych zwyczajnych, których postać zdefiniowana jest w formie zbioru wykonywalnego typu exe lub com. Ponieważ uruchomienie tego zbioru ma sens jedynie wtedy, gdy uruchominy jest program główny Turbo Wiener™, to by zabezpieczyć go przez przypadkowym uruchomieniem, program Turbo Wiener™ zmienia jesgo rozszerzenia na sys.

Rozwiązanie to jest proste i efektywne. Pozwala na oszczędną gospodarkę pamięcią operacyjną, gdyż informacja o tym systemie jest niezbędna jedynie w kilku miejscach programu Turbo Wiener™. Przyjęta metoda umożliwia pełną swobodę wyboru języka programowania do opisu systemu. Sam opis systemu jest komplikowany oddzielnie i nie jest konieczna powtórna kompilacja programu Turbo Wiener™.

Rozwiązanie równań różniczkowych zwyczajnych opisujących system modelowany, dokonywane jest w programie Turbo Wiener™ za pomocą metody Braytona-Gustavsona=Hatchela (M. M. Stabrowski [51]).

Ponieważ modele Laguerre'a zawierają wielomiany i funkcje Laguerre'a, to przy opracowywaniu programu Turbo Wiener™, koniecznym było określenie metody wyznaczania wartości wielomianów oraz funkcji Laguerre'a. Ze względu na dużą złożoność obliczeniową algorytmów identyfikacji systemów, do wyznaczania wartości tych wielomianów zastosowano regułę trójczłonową, która zapewnia maksymalną szybkość obliczeń (J. Jankowska, M. Jankowski [30], s. 93). Z kolei przy obliczaniu splotów funkcji Laguerre'a z wymuszeniami zastosowano algorytm FFT, dla którego 256 wartości próbek funkcji Laguerr'a z wymuszeniami zastosowano algorytm FFT, dla którego 256 wartości próbek funkcji Laguerr'a wyznaczane jest za pomocą kwadratury Gaussa-Legendre'a (patrz A. Ralston [41], s. 91).

Z przedstawionej wcześniej teorii wynika, że przy wyznaczaniu modelu systemu, niezbędnym jest rozwiązanie układu równań liniowych. W omawianym programie zastosowano algorytm QR Francisa (J. Stoer, R. Bulirsch [52], s. 382 oraz G. Dahlquist, A. Björck [25], s. 197) z podwójnym ulepszeniem (G. Dahlquist, A. Björck [25], s. 200). Iloczyny skalarne występujące w pojawiających się tu układach równań normalnych, zostały wyznaczone za pomocą złożonej kwadratuey Newtona-Cotesa drugiego rzędu, nazywanej złożonym wzorem trapezów (S. Stoer, R. Bulirsch [52], s. 128).

Jako źródło szumu białego wykorzystano w programie Turbo Wiener™ multiplikatywny generator liczb pseudolosowych zaproponowany przez O. Tausski (D. E. Knuth [32], s. 112), dla którego n-ta liczba pseudolosoa

$$r_n := 5^{15} r_{n-1} (\mathrm{mod} 2^{35}) .$$

Generator ten wyróżnia się spośród innych generatorów liczb pseudolosowych, bardzo dobrymi parametrami widmowymi, związanymi z gęstością widmową mocy (D. E. Knuth [32], s. 105).

8. PRZEBIEG OBLICZEŃ

Za pomocą programu Turbo Wiener™ wyznaczono model Laguerre'a dla *systemu inercyjnego pierwszego rzędu* opisanego za pomocą równania:

$$\tau \frac{dv}{dt} + v = ku,$$

gdzie k jest *współczynnikiem wzmocnienia* zdefginiowanym jako iloraz odpowiedzi v do wymuszenia u w stanie ustalonym, a τ jest *stałą czasową*, oraz *systemu oscylacyjnego* opisanego za pomocą równania:

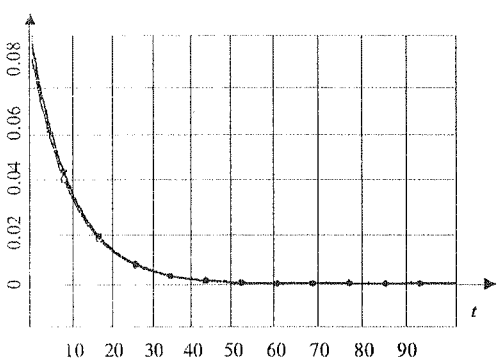
$$T_0^2 \frac{d^2v}{dt^2} + 2\xi T_0 \frac{dv}{dt} + v = ku,$$

gdzie k jest *współczynnikiem wzmocnienia* zdefiniowanym tak jak poprzednio, T_0 jest *okresem drgań własnych nie tłumionych*, a $\xi \in [0,1]$ jest *względnym współczynnikiem tłumienia*. Parametry systemu intercyjnego pierwszego rzędu wynosiły tu:

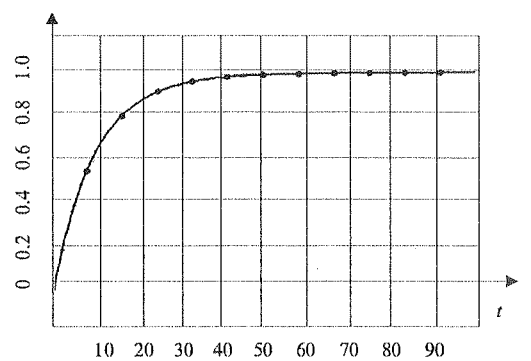
$$k = 1 \text{ oraz } \tau = 10\,\text{s}.$$

Z kolei parametry systemu oscylacyjnego były tu:

$$k = 1,\ T_0 = 10\,\text{s} \text{ oraz } \xi = 0.5.$$



Rys. 8. 1. Odpowiedź impulsowa systemu inercyjnego pierwszego rzędu ° i jego modelu •.



Rys. 8. 2. Odpowiedź skokowa systemu inercyjnego pierwszego rzędu ° i jego modelu •.

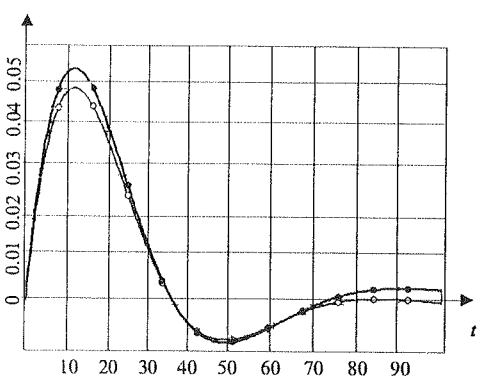
Zbiory z opisem tych systemów mają nazwy inertlor.sys i oscilat.sys, a ich parametry zdefinowane są w zbiorach inertlot.dat i oscilat.dat. Utworzone zbiory z wyniakmi mają nazyw odpowiednio inertlor.mod i oscilat.mod.

Modele systemu inercyjnego pierwszego rzędu i systemu oscylacyjnego wyznaczano dla różnych wartości współczynnika skali czasu c, stopnia systemu Laguerre'a $k-1$ oraz współczynnika widmowego. Współczynnik widmowy określa

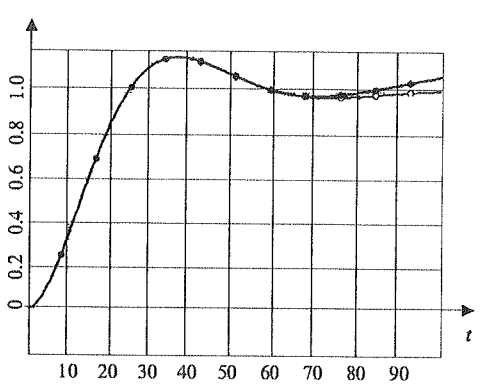
krotność próbek szumu białego i powinien być potęgą liczby 2. Okazało się, że suboptymalna wartość współczynnika skali czasu

$$c = 0.1\frac{1}{s},$$

a współczynnika widmowego wynosiła 32. Ponadto maksymalna wartość szumu wynosiła tu 1, a czas uśredniania wynosiła 100 s. Model systemu untercyjnego utworzony jest z 0 funkcji Laguerre'a, a w przypadku systemu oscyalcyjnego korektor wykorzystuje 8 kolejnych funkcji Laguerre'a.



Rys. 8. 3. Odpowiedź impulsowa systemu oscylacyjnego ° i jego modelu •.



Rys. 8. 4. Odpowiedź skokowa systemu oscylacyjnego ° i jego modelu •.

Odpowiedź impulsowa systemu intercyjnego pierwszego rzędu i jego modelu przedstawiono na rys. 8.1, a odpowiedź skokowa tego systemu i jego modelu przedstawiona została na rys. 8.2. Z kolei odpowiedź impulsowa systemu oscylacyjnego i jego modelu przedstawiono na rys. 8.3, a odpowiedź skokowa tego systemu i jego modelu przedstawiona została na rys. 8.4.

9. PODSUMOWANIE

Za pomocą programu Turbo Wiener™ wyznaczono model Laguerre'a dla systemu inercyjnego i oscylacyjnego. Otrzymane wyniki są godne szczególnej uwagi. Uzyskane bowiem wyniki bardzo dobrze opisują odpowiedzi impulsowe, jak i skokowe systemów intercyjnych i oscylacyjnych. W przypadku systemu intercyjnego stopień przybliżającego wielomianu Laguerre'a wynosił 0, a w przypadku systemu oscylacyjnego stopień ten wynosił 7.

Autor artykułu pracuje obecnie nad nowym typem operatorów, nazywanych roboczo operatorami sklejanymi, które wykorzystują funkcje sklejane ([5], [7], [28] i [49]). Operatory te powinny odegrać podobną rolę, jaką odegrały kiedyś operatory Volterry i Wienera w teorii nieliniowych systemów stacjonarnych przyczynowych ([39], [40] oraz [42]—[48]).

zało się, że

tość szumu
ntercyjnego
go korektor

stemu oscyla-
•.

go modelu
go modelu
oscylacyj-
go systemu

ierre'a dla
lnej uwagi.
ak i skoko-
tercyjnego
u systemu

azywanych
5], [7], [28]
operatory
czynowych

BIBLIOGRAFIA

1. W. M. A m i e r b a j e w, N. A. U t i m b a j e w: *Czislenyj analiz łagierrowskowo spiektra.* Nauka, Ałma-Ata, 1982.
2. J. S. B e n d a t, A. G. P i e r s o l (tłum. a ang. J. D u d z i e w i c z, R. B i a ł o b r z e s k i): *Metody analizy i pomiaru sygnałów losowych,* Biblioteka Naukowa Inżynieria, Warszawa, PWN, 1976.
3. P. B i l l i n g s l e y (tłum. z ang. K. K i z e w e t e r, J. E. R o g u l s k i): *Prawdopodobieństwo i miara.* Warszawa, PWN, 1987.
4. G. B i r k h o f f, T. C. B a r t e e (tłum. z ang. M. S z y m a ń s k a): *Współczesna algebra stosowana,* Matematyka dla politechniki, Warszawa, PWN, 1987.
5. C. de B o o r: *Practical to Splines.* Applied Mathematical Science, Vol. 27 New York, Springer-Verlag, 1978.
6. R. K. B r a y t o n, F. G. G u s t a v s o n, G. D. H a t c h e l: *A new efficient algorithm for solving differential-algebraic systems using implicit backward differetiation formulas.* Proc. of IEEE, Vol. 60 (1), 1972 pp. 98 – 108.
7. G. C i e s i e l s k i: *Kształtowanie statystycznych charakterystyk czujników do pomiaru wielkości nieelektrycznych za pomocą przybliżeń krzywikowymi.* Rozprawa doktorska, Instytut Podstaw Elektrotechniki, Politechnika Łódzka, Łódź, 1982.
8. G. C i e s i e l s k i: *Aproksymacja średniokwadratowa operatorów nieliniowych opisujących wielowymiarowe systemy pomiarowe.* XXIII Międzyuczelniana Konferencja Metrologów, Politechnika Warszawska, Warszawa, 1991 (referat nie opublikowany).
9. G. c i e s i e l s k i: *Modele Wienera wielowymiarowych systemów pomiarowych opisanych za pomocą operatorów nieliniowych. modelowanie i symulacja systemów pomiarowych.* Materiały Sympozjum, AGH, Kraków, 1992, ss. 41 – 50.
10. G. C i e s i e l s k i: *Operatorowe modele wielowymiarowych systemów pomiarowych.* Seminarium Naukowe Komisji Kształcenia Komitetu Metrologii i Aparatury Naukowej Polskiej Akademii Nauk, 1994 (referat nie opublikowany).
11. G. C i e s i e l s k i: *Identyfikacja ogólnej struktury wielowymiarowych systemów pomiarowych. modelowanie i symulacja systemów pomiarowych.* Materiały sympozjum, AGH, Kraków, 1994, ss. 17 – 23.
12. G. C i e s i e l s k i: *Identyfikacja ogólnej struktury wielowymiarowych systemów stacjonarnych,* Kwartalnik Elektroniki i Telekomunikacji, T. 40, z. 3, 1994, ss. 419 – 428.
13. G. C i e s i e l s k i: *modelowanie wielowymiarowych systemów stacjonarnych za pomocą splotów.* Kwartalnik Elektroniki i Telekomunikacji, T. 40, z. 3, 1994, ss. 429 – 448.
14. G. C i e s i e l s k i: *Modelowanie i korekcja wielowymiarowych stacjonarnych systemów pomiarowych za pomocą operatorów nieliniowych.* Rozprawa habilitacyjna, Politechnika Łódzka, Zeszyty Naukowe, Nr 699, Łódź, 1994.
15. G. C i e s i e l s k i: *Identyfikacja szczegółowej struktury wielowymiarowych systemów stacjonarnych.* Kwartalnik Elektroniki i Telekomunikacji, T. 40, z. 3, 1995, ss. 305 – 319.
16. G. C i e s i e l s k i: *Modelowanie wielowymiarowych systemów stacjonarnych przyczynowych.* Kwartalnik Elektroniki i Telekomunikacji, T. 41, z. 3, 1995, ss. 321 – 335.
17. G. C i e s i e l s k i: *Modelowanie wielowymiarowych liniowych systemów stacjonarnych w systemach pomiarowych. Modelowanie i symulacja systemów pomiarowych.* Materiały Sympozjum, AGH, Kraków, 1996, ss. 28 – 37.
18. G. C i e s i e l s k i: *Identyfikacja szczegółowej struktury wielowymiarowych systemów stacjonarnych w systemach pomiarowych. Modelowanie i symulacja systemów pomiarowych.* Materiały Sympozjum, AGH, Kraków, 1996, ss. 38 – 42.
19. G. C i e s i e l s k i: *Zastosowanie modeli wienera w systemach pomiarowych. systemy pomiarowe w badaniach naukowych i w przemyśle.* Materiały konferencyjne, Wyższa Szkoła Inżynierska, Zielona Góra, 1996, ss. 43 – 52.
20. G. C i e s i e l s k i: *Modelowanie wielowymiarowych systemów stacjonarnych w systemach pomiarowych, Modelowanie i symulacja systemów pomiarowych.* Materiały Sympozjum, AGH, Kraków, 1997, ss. 129 – 142.

21. G. Ciesielski: *Korekcja wielowymiarowych systemów stacjonarnych w systemach pomiarowych. modelkowanie i symulacja systemów pomiarowych.* Materiały Sympozjum, AGH, Kraków, 1997, ss. 143–150.
22. G. Ciesielski: *Korekcja wielowymiarowych systemów stacjonarnych przyczynowych.* Kwartalnik Elektroniki i Telekomunikacji (artykuł złożony do druku).
23. G. Ciesielski: *Korekcja liniowych systemów stacjonarnych za pomocą systemów Laguerre'a.* Kwart. Elektr. i Telekom. (artykuł złożony dso druku).
24. G. Ciesielski: *General structure of multidimensional time-invariant systems.* Communications on Pure and Applied Mathematics. John Wiley & Sons, Inc. (artykuł w przygotowaniu).
25. G. Dahlquist, Å. Björck (tłum. z ang. S. Paszkowski): *Metody numeryczne.* Warszawa, PWN, 1983.
26. P. Eykhof (tłum. z ang. A Bauer): *Identyfikacja w układach dynamicznych.* Biblioteka Naukowa Inżynieria, Warszawa, PWN, 1980.
27. R. J. P. de Figueiredo, T. A. W. Dwyer: *A best approximation framework and implementation for simulation of large-scale nonlinear systems.* IEEE Trans. Circuits Syst., Vol. CAS-27, No 11, pp. 1005–1014
28. R. J. de Figueiredo: *Nonlinear circuits and systems applications of functional splines.* Proc. IEEE, 1982, pp. 1245–1247.
29. S. Gładysz: *Wstęp do topologii.* Matematyka dla politechniki, Warszawa, PWN, 1981.
30. J. Jankowska, M. Jankowski: *Przegląd metod i algorytmów numerycznych.* T. 1, Warszawa, WNT, 1981.
31. J. Jaworski, R. Morawski, J. Olędzki: *Wstęp do metrologii i techniki eksperymentu.* Warszawa, WNT. 1992.
32. D. E. Knuth: *The Art of Computer Programming.* Vol. 2, Massachusetts, Addison-Wesley, 1969.
33. J. Kudrewicz: *Częstotliwościowe metody w teorii nieliniowych układów dynamicznych.* Warszawa, WNT, 1970.
34. K. Kuratowski: *Wstęp do teorii mnogości i topologii.* Biblioteka matematyczna, T. 9, Warszawa, PWN, 1980.
35. F. Morrison: *Sztuka modelowania układów dynamicznych, deterministycznych, chaotycznych, stochastycznych.* Warszawa, WNT. 1996.
36. J. Musielak: *Wstęp do analizy funkcjonalnej.* Warszawa, PWN, 1989.
37. J. Osiowski: *Zarys rachunku operatorowego. Teoria i zastosowanie w elektrotechnice.* Warszawa, WNT, 1981.
38. A. Papoulis: *Prawdopodobieństwo, zmienne losowe i procesy stochatyczne.* Warszawa, WNT, 1972.
39. J. Park, I. W. Sandberg: *Criteria for the approximation of nonlinear systems.* IEEE Trans. Circuits Syst.-I: Fundamental Theory and Applications, Vol. 39, No 8, 1992, pp. 673–676.
40. K. A. Pupkow, W. I. Kapalin, A. S. Juszczenko: *Funkcjnalnyje rjady w tjeorii nieliniejnych sistjem.* Moskwa, Nauka, 1976.
41. A. Ralston (tłum. z ang. R. Zuber): *Wstęp do analizy numerycznej.* Warszawa, PWN, 1983.
42. I. W. Sandberg: *Criteria for the respondense of nonlinear to be L-asymptotic periodic.* Bell Syst. Tech. J., Vol. 60, No 10, 1981, pp. 2359–2371.
43. I. W. Sandberg: *Series expansion for nonlinear systems.* Proc. IEEE, 1982, pp. 110–113.
44. I. W. Sandberg: *Expansion for nonlinear systems.* Bell Syst. Tech. J., Vol. 61, No 2, pp. 159–199.
45. I. W. Sandberg: *Volterra expansion for time-verying systems* Bell Syst. Tech. J., Vol. 61, No 2, 1982, pp. 201–225.
46. I. W. Sandberg: *Multidimensional nonlinear systems and structure theorems.* J. Circuits, syst. and Computers, Vol. 2, No 4, 1992, pp. 383–388.
47. I. W. Sandberg: *Approximately-finite memory and input-output maps.* IEEE Trans. Circuits syst.-I: Fundamental Theory and Applications, Vol. 39, No 7, 1992, pp. 549–556.
48. M. Schetzen: *The Volterra and Wiener Theories of Nonlinear Systems.* New York, John Wiley & Sons, 1980.
49. L. L. Schumaker: *Spline Functions. Basic Theory..* New York, John Wiley & Sons, 1981.

50. T. Söderström, P. Stoica: *Identyfikacja systemów*. Warszawa, PWN, 1997.
51. M. M. Stabrowski: *Pivoted block solvers for large banded linear equation systems*. communicat. In Num. Meth. In Eng., Vol. 3, 1997, pp. 407–415.
52. JH. Stoer, R. Bulirsch (tłum. z ang. J. Cytowski): *Wstęp na analizy numerycznej*. Warszawa, PWN, 1987.
53. J. Szabatin: *Podstawy teorii sygnałów*. Warszawa, WNT, 1982.
54. B. Widrow, S. D. Stearns: *Adaptive Signal Processing*. New Jersey, Prentice-Hall, Inc, 1985.
55. A. Wojnar: *Teoria sygnałów*. Podręczniki akademickie EIT, Warszawa, WNT 1980.
56. R. R. Yager, D. P. Filev: *Podstawy modelowania i sterowania rozmytego*. Warszawa, WNT, 1995.

G. CIESIELSKI

MODELING OF LINEAR AND TIME-INVARIANT SYSTEMS BY LAGUERRE SYSTEMS

Summary

The problem of structure and parameter identifications of models of linear time-invariant system is presented in this paper. The structure of the Laguerre models is applied to they describes. Presented theory invoves from uniform general operator's theory of systems linear as well as nonlinear. Uniform general operator's theory of systems has evaluated by author of this paper, from several years ([7]–[24]). In this paper, the Turbo Wiener™ program, designed and evaluated by authors for MS Windows operational system, is presented. This program lets to calculate the models and correctors of time-invariant systems, which are linear nonlinear. There is applied uniform general operator's theory of systems. This paper presents too the results of computation by use of program Turbo Wiener ™ for typical linear time-invariant systems.

Key words: universal functional theory of systems, modeling, structure identification, parametric identification, linear systems, time-invariant systems, causal systems, operators, convolutions, stationary space, convolution space, causal space, program Turbo Wiener™.

A
torów
re'a.
zarów
rozwi
progr
MS V
zarów
ogóln
nych
stacjo

Słowa
cja st
systen
przest

Artyku
metryczną
Morawski
Przedstaw
zarówno
temów jes

* Pracę w
Badań Nauk

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2000, 46, z. 2, ss. 149–161

Korekcja liniowych systemów stacjonarnych za pomocą systemów Laguerre'a*

GRZEGORZ CIESIELSKI

Instytut Elektroniki Teoretycznej, Metrologii i Materiałoznawstwa, Politechnika Łódzka

Otrzymano 1999.05.10
Autoryzowano 2000.02.24

Artykuł omawia zagadnienia identyfikacji strukturalnej i parametrycznej modeli korektorów liniowych systemów stacjonarnych. Do ich opisu przyjęto strukturę modeli Laguerre'a. Przedstawiona tu teoria wynika z jednolitej ogólnej teorii operatorowej systemów, zarówno liniowych, jaki i nieliniowych. Jednolita ogólnej teorii operatoraowa systemów jest rozwijana przez autora artykułu od szeregu lat ([7]–[24]). W artykule omówiono również program Turbo Wiener™, opracowany i rozwijany przez autora dla systemu operacyjnego MS Windows, który pozwala wyznaczyć modele i korektory systemów stacjonarnych, zarówno liniowych, jak i nieliniowych, wykorzystując przy tym wspomnianą jednolitą ogólną teorię operatorową systemów. Przedstaawiono tu również wyniki obliczeń otrzymanych za pomocą programu Turbo Wiener™, dla korektorów typowych liniowych systemów stacjonarnych.

Słowa kluczowe: ogólna funkcjonalna teoria systemów, korekcja, modelowanie, identyfikacja strukturalna, identyfikacja parametryczna, systemy wielowymiarowe, systemy liniowe, systemy stacjonarne, systemy przyczynowe, operatory, sploty, przestrzeń stacjonarna, przestrzeń splotowa, przestrzeń przyczynowa, program Turbo Wiener™.

1. WSTĘP

Artykuł omawia zagadnienia korekcji, a więc identyfikację strukturalną i parametryczną modeli korektorów liniowych systemów stacjonarnych (J. Jaworski, R. Morawski, J. Olędzki [31]). Do ich opisu przyjęto strukturę modeli Laguerre'a. Przedstawiona tu teoria wynika z jednolitej ogólnej teorii operatorowej systemów, zarówno liniowych, jak i nieliniowych. Jednolita ogólna teoria operatorawa systemów jest rozwijana przez autora artykułu od szeregu lat ([7]–[24]). W artykule

* Pracę wykonano w ramach projektu badawczego nr 8 T10C 031 09 finansowanego przez Komitet Badań Naukowych w latach 1995–1997.

omówiono również program Turbo Wiener™, opracowany i rozwijany przez autora dla systemu operacyjnego MS Windows. Program ten pozwala wyznaczać modele i korektory systemów stacjonarnych, zarówno liniowych, jak i nieliniowych. Wykorzystuje on przy tym wspomianą jednolitą ogólną teorię operatorową systemów. Program Turbo Wiener™ został w niniejszym artykule omówiony jedynie w części dotyczącej modelowania korektorów liniowych systemów stacjonarnych. Przedstawiono tu również wyniki obliczeń otrzymanych za pomocą programu Turbo Wiener™, dla korektorów typowych liniowych systemów stacjonarnych, inercyjnego i oscylacyjnego.

W artykule wykorzystywane są PRELIMINARIA MATEMETYCZNE z poprzedniego artykułu (G. Ciesielski [23]).

2. IDENTYFIKACJA STRUKTURALNA

Zadanie korekcji dowolnego *systemu* ϕ ma na celu wyznaczenie matematycznego modelu korektora tego systemu. Zadanie to składa się z *identyfikacji strukturalnej* oraz *identyfikacji parametrycznej* modelu (J. Jaworski, R. Morawski, J. Olędzki [31]) korektora. Zadanie identyfikacji strukturalenj daje w wyniku strukturę poszukiwanego korektora, a zadanie identyfikacji parametrycznej pokazuje jak wyznaczać parametry struktury będącej wynikiem identyfikacji strukturalnej.

Omówimy teraz metody wyznaczania parametrów struktury będącej wynikiem identyfikacji strukturalnej oraz identyfikacji parametrycznej, o których była mowa w poprzednim artykule (G. Ciesielski [23]). Otrzymane wyniki wykorzystamy do korekcji systemu ϕ.

Z udowodnionego w artykule [23] twierdzenia o identyfikacji strukturalnej liniowych systemów stacjonarnych wynika, że każdy linowy ciągły system stacjonarny przyczynowy $\phi: \mathbf{C}_{\mathbf{R}\infty+} \mapsto \mathbf{C}_{\mathbf{R}\infty+}$ można opisać za pomocą splotu funkcji, a zatem

$$\phi(u) = g^* u ,$$

gdzie g jest odpowiedzią impulsową tego systemu. Odpowiedź impulsowa g jest całkowalna, a gdy ponadto odpowiedź g jest istotnie ograniczona, to jest ona również całkowalna w kwadracie. Jest to bardzo ważne twierdzenie należące w analizie funkcjonalnej do grupy twierdzeń o reprezentacjach odwzorowań. Twierdzenia należące do tej grupy dotyczą zadania identyfikacji strukturalnej różnego rodzaju odwzorowań, również korektorów.

Zauważmy, że założenie w twierdzeniu tym, ciągłości na $\mathbf{R}$ odpowiedzi systemu ϕ nie jest zbyt drastyczne, gdyż każdy fizycznie realizowany system ma skończone pasmo przenoszenia, a stąd wynika ciągłość na $\mathbf{R}$ jego odpowiedzi. Podobnie założenie ograniczoności na $\mathbf{R}$ odpowiedzi systemu ϕ wynika ze skończonej mocy układów zasilających ten system. Zauważmy tu również, że zgodnie z definicją stacjonarności operatora, sformułowaną w [23], przestrzeń wymuszeń operatora stacjonarnego musi zawierać wszystkie ujemne ($\sigma \leqslant 0$) przesunięcia funkcyjne swoich

elementów. Przestrzeń $\mathbf{C}_{R\infty+}$ ma tą właściwość, co pozwala nam definiować na niej operatory stacjonarne.

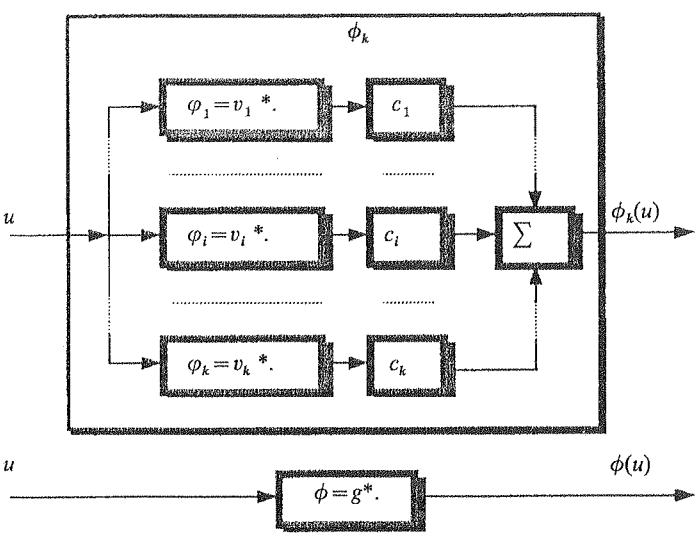
Zajmiemy się teraz identyfikacją strukturalną korektorów liniowych ciągłych systemów stacjonarnych przyczynowych $\phi:\mathbf{C}_{R\infty+}\mapsto\mathbf{C}_{R\infty+}$ o całkowalnych w kwadracie odpowiedziach impulsowych. Z udowodnionego w artykule [23] twierdzenia o identyfikacji strukturalnej liniowych systemów stacjonarnych wynika, że każdy taki system ϕ można przybliżyć za pomocą kombinacji liniowej liniowych systemów stacjonarnych przyczynowych $\phi_{[k]}:=(\phi_i)_1^k$, których odpowiedzi impulsowe $(v_i)_1^k=:v_{[k]}$ tworzą układ liniowo niezależny w $\mathbf{L}_{R2+}$. Zauważmy bowiem, że $\forall u\in\mathbf{C}_{R\infty+}$

$$\phi_k(u):=g_k*u=(c_{[k]}^T v_{[k]})*u=c_{[k]}^T(v_{[k]}*u):=c_{[k]}^T(v_i*u)_1^k,$$

gdzie

$$c_{[k]}:=(c_i\in\mathbf{R})_1^k=G^{-1}(v_{[k]})\langle g,v_{[k]}\rangle:=G^{-1}(v_{[k]})(\langle g,v_i\rangle)_1^k.$$

Przybliżenie ϕ_k można więc schematycznie przedsawić tak jak to zostało pokazane na rys. 2. 1. Przybliżenie to jest też przybliżeniem poszukiwanych korektorów liniowych ciągłych systemów stacjonarnych przyczynowych $\phi:\mathbf{C}_{R\infty+}\mapsto\mathbf{C}_{R\infty+}$ o całkowalnych w kwadracie odpowiedziach impulsowych. Identyfikacja parametryczna poszkiwanego korektora sprowadza się więc do postępowania mającego na celu wyznaczenie współczynników kombinacji liniowej, które składają się na wektor $C_{[k]}$. Wymaga to jak widać wyznaczenia odpowiednich iloczynów skalarnych w $\mathbf{L}_{R2}$.



Rys. 2.1. Struktura przybliżenia ϕ_k liniowego ciągłego systemu stacjonarnego przyczynowego ϕ: $C_{R\infty+}\mapsto C_{R\infty+}$ o całkowalnej w kwadracie odpowiedzi impulsowej.

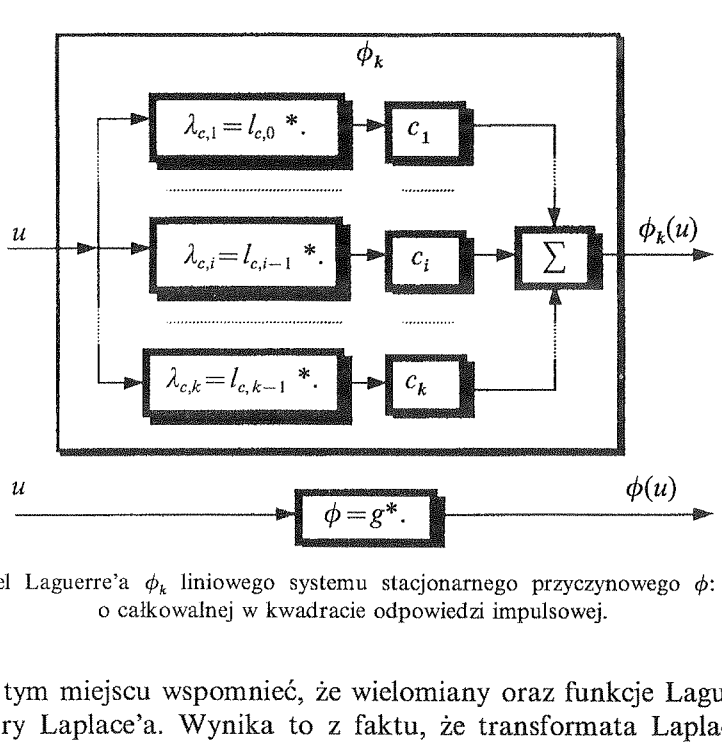
Jeżeli system ϕ jest dany w postaci analitycznej, to zadanie identyfikacji parametrycznej jest dość prostym problemem obliczeniowym, polegającym na rozwiązaniu typowego zadania aproksymacji średniokwadratowej. Zajmiemy się zatem

dalej systemami danymi w postaci nieanalitycznej, a ściślej, jedną z najważniejszych reprezentacji tej postaci, dla której mamy dostępne jedynie wejście i wyjście systemu ϕ. Twierdzenie o identyfikacji parametrycznj (G. Ciesielski [23]) pozwala rozwiązać zadanie identyfikacji parametrycznej.

Pojawiający się tu problem polega na wyznaczaniu postaci układu liniowo niezależnego $v_{[k]} := (v_i)_1^k$ w $\mathbf{L}_{\mathbf{R}2+}$, o którym była mowa w twierdzeniu o identyfikacji strukturalnej (G. Ciesielski [23]). Można wykazać, że układ funkcji Laguerre'a może być użyty do budowy układu liniowo niezależnego $v_{[k]} := (v_i)_1^k$ z twierdzenia o identyfikacji strukturalnej. Wynika to z faktu, że każdy układ ortogonalny, a zatem i ortonormalny, jest liniowo niezależny (J. Musielak [36], s. 70).

Definicja wielomianów i funkcji Laguerre'a można znaleźć w [23].

Wynikającą teraz strukturę przybliżenia ϕ_k liniowego ciągłego systemu stacjonarnego przyczynowego $\phi: \mathbf{C}_{\mathbf{R}\infty+} \mapsto \mathbf{C}_{\mathbf{R}\infty+}$ o całkowalnej w kwadracie odpowiedzi impulsowej, pokazano na rys. 2.2 tegoż artykułu. Przybliżenie ϕ_k o tej postaci nazywane jest *modelem Laguerre'a*. Zauważmy, że w modelu tym układ liniowo niezależny $v_{[k]} := (v_i)_1^k = (l_{c,i-1})_1^k$, a zatem model Laguerre'a jest kombinacją liniową systemów stacjonarnych przyczynowych $\lambda_{[k]} := (\lambda_{c,i})_1^k$ nazywanych *systemami Laguerre'a* których odpwiedzi impulsowe $(l_{c,i-1})_1^k =: v_{[k]}$ tworzą uklad liniowo niezależny w $\mathbf{L}_{\mathbf{R}2+}$.



Rys. 2.2. Model Laguerre'a ϕ_k liniowego systemu stacjonarnego przyczynowego ϕ: $C_{R\infty+} \mapsto C_{R\infty+}$ o całkowalnej w kwadracie odpowiedzi impulsowej.

Warto w tym miejscu wspomnieć, że wielomiany oraz funkcje Laguerre'a mają transformatory Laplace'a. Wynika to z faktu, że transformata Laplace'a funkcji Laguerre'a oznaczane przez

$$[\mathscr{L}(1_{c,n})](s) = \frac{\sqrt{2c}}{c+s}\left(\frac{c-s}{c+s}\right)^n$$

(M. Schetzen [48], s. 352), a stąd i z twierdzenia o przesunięciu w dziedzinie zespolonej (J. Osiowski [37], s. 138) mamy, że jednostronna transformata Laplace'a wielomianu Laguerre'a

$$[\mathscr{L}(L_{c,n})](s)=[\mathscr{L}(l_{c,n}\exp(c\cdot))](s)=\frac{\sqrt{2c}}{s}\left(\frac{2c-s}{s}\right)^n.$$

Z liniowości przekształcenia Laplace'a wynika zatem, że funkcje Laguerre'a umożliwiają numeryczne wyznaczanie transformat Laplace'a dowolnych funkcji należących do przestrzeni $\mathbf{L}_{\mathbf{R}2+}$, gdyż stanowią dla niej bezę ortonormalną (patrz też W. M. Amierbajew, N. A. Utiembajew [1], s. 9). Wynika stąd, iż możemy też wyznaczać numerycznie transformaty Laplece'a poszukiwanych korektorów dla liniowych systemów stacjonarnych przyczynowych.

3. IDENTYFIKACJA PARAMETRYCZNA

Załóżmy, że dane są dwa systemy, *system korygowany* $\phi:\mathbf{C}_{\mathbf{R}\infty+}\mapsto\mathbf{C}_{\mathbf{R}\infty+}$ oraz *system wzorowany* $\eta:\mathbf{C}_{\mathbf{R}\infty+}\mapsto\mathbf{C}_{\mathbf{R}\infty+}$. Załóżmy ponadto, że systemy te są liniowymi ciągłymi systemami stacjonarnymi przyczynowymi o całkowalnych w kwadracie odpowiedziach impulsowych. *Zadania korekcji systemu* ϕ polega na takim doborze korektora γ_k by odpowiedź impulsowa $g_{\gamma k}*g_\phi$ systemu $\gamma_k\circ\phi$ minimalizowała wyrażenie

$$\| g_\eta - g_{\gamma k}*g_\phi \|_2,$$

w którym $\|\cdot\|_2$ jest normą w $\mathbf{L}_{\mathbf{R}2}$, a g_η jest odpowiedzią impulsową systemu wzorcowego η. Tak sformułowane zadanie korekcji pokazane zostało schematycznie na rys. 2.1. (a). Z rysunku tego wynika, że zadanie to sprowadza się do wyznaczenia współczynników kombinacji liniowej

$$c:=(c_i\in\mathbf{R})_1^k$$

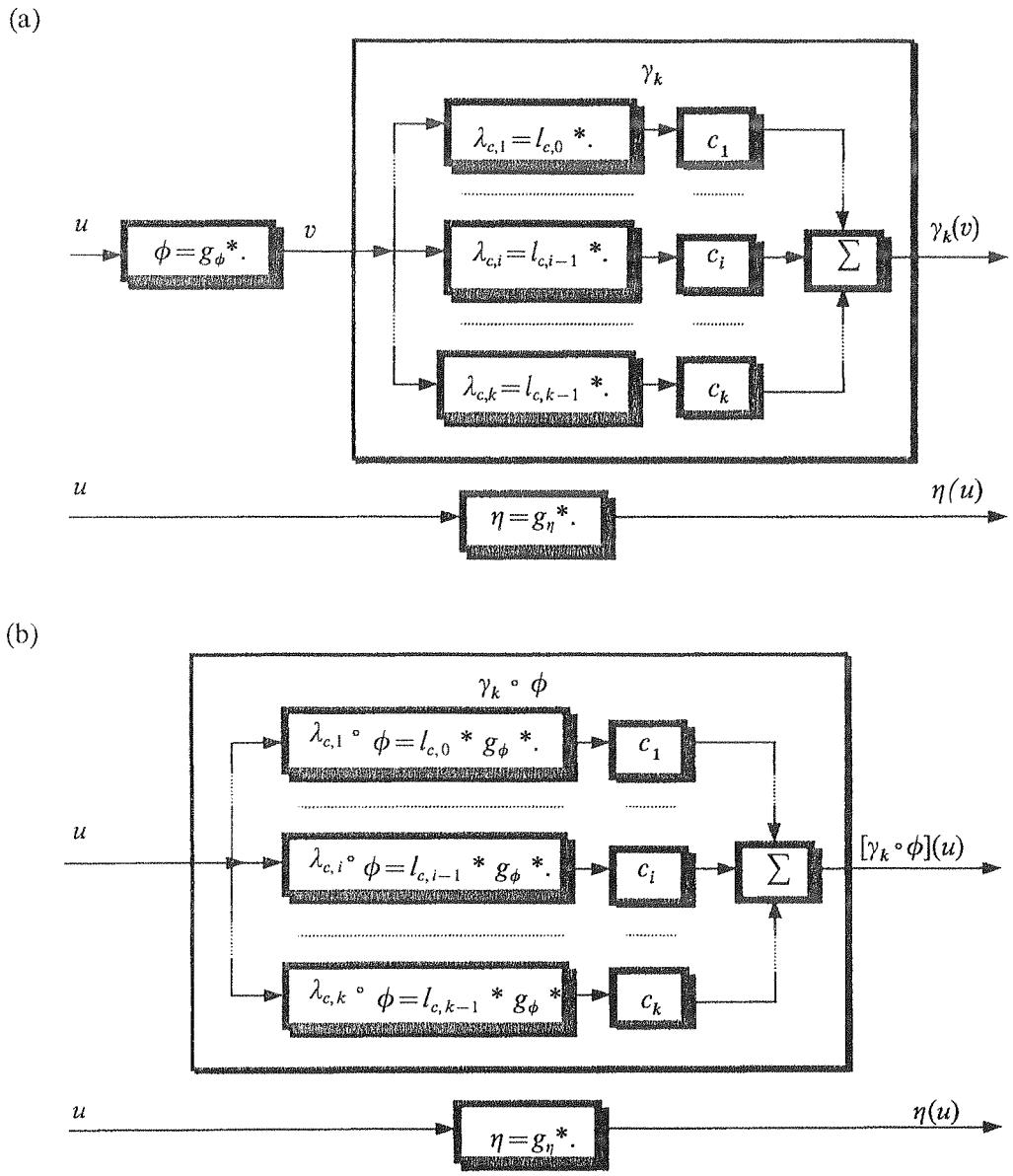
dla układu liniowo niezależnego $(l_{c,i-1})_1^k$. Zauważmy, że przyjęcie takiej struktury korektora jest wynikiem wcześniejszych rozważań dotyczących identyfikacji strukturalnej (G. Ciesielski [23]) oraz wyboru bazy ortonormalnej w przestrzeni $\mathbf{L}_{\mathbf{R}2+}$ (G. Ciesielski [23]).

Przedstawiona tu *postać zadania korekcji* jest *niekonstruktywna*, gdyż nie wynika z niej jasno sposób wyznaczania współczynników kombinacji liniowej składających się na wektor c. Dlatego też na rys. 3.1. (b) przedstawiono *zadanie korekcji w postaci konstruktywnej*. Obie te postacie są oczywiście równoważne, jednak z postaci konstruktywnej wynika, że zadanie korekcji można sprowadzić do zadania aproksymacji średniokwadratowej odpowiedzi impulsowej g_η systemu wzorcowego za pomocą kombinacji liniowej odpowiedzi impulsowych

$$v_{[k]}:=(v_i)_1^k:=(l_{c,i-1}*g_\phi)_1^k$$

liniowych systemów stacjonarnych przyczynowych

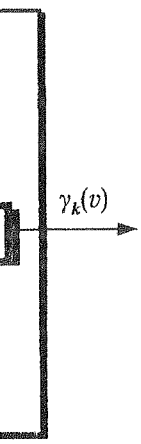
$$\lambda_{c[k]}\circ\phi=(\lambda_{c,i}\circ\phi)_1^k.$$



Rys. 3.1. Niekonstruktywna (a) i konstruktywna (b) postać zadania korekcji systemu ϕ: $C_{R\infty+} \mapsto C_{R\infty+}$, dla systemu wzorcowego η: $C_{R\infty+} \mapsto C_{R\infty+}$. System ϕ oraz η są liniowymi systemami stacjonarnymi przyczynowymi o całkowalnych w kwadracie odpowiedziach impulsowych.

Jeżeli układ $v_{[k]}$ jest liniowo niezależny, co na ogół jest spełnione, to dla wyznaczenia wektora $c_{[k]}$ możemy wykorzystać teorię z [23], z której wynika, że

$$c_{[k]} := (c_i)_1^k = G^{-1}(v_{[k]})\langle g_\eta, v_{[k]}\rangle = (A((v_{[k]}*x)(v_{[k]}*x)^T))^{-1}A((g_\eta*x)(v_{[k]}*x)) .$$



$\eta(u)$

$\gamma_k \circ \phi](u)$

$\eta(u)$

$_{R\infty+} \mapsto C_{R\infty+}$, dla
i stacjonarnymi
.

wyznaczenia

$_k]*x))$.

4. PROGRAM TURBO WIENER™ DO WYZNACZANIA KOREKTORÓW LINIOWYCH SYSTEMÓW STACJONARNYCH

Jakość oraz efektywność przedstawionych wcześniej metod wyznaczania parametrów (identyfikacji parametrycznej) modeli korektorów liniowych ciągłych systemów stacjonarnych przyczynowych o całkowalnej w kwadracie odpowiedzi impulsowej, można przebadać za pomocą programu Turbo Wiener™.

Jednym z problemów jakie należało rozwiązać w trakcie pisania programu Turbo Wiener™, był sposób opisu systemu korygowanego i wzorcowego. W programie Turbo Wiener™ założono, że systemy te są opisane za pomocą liniowych równań różniczkowych zwyczajnych, których postać zdefiniowana jest w formie zbiorów wykonywalnych typu exe lub com. Ponieważ uruchmienie tych zbiorów ma sens jedynie wtedy, gdy uruchominy jest program główny Turbo Wiener™, to by zabezpieczyć je przez przypadkowym uruchomieniem program Turbo Wiener™ zmienia ich rozszerzenia na sys.

Rozwiązanie to jest proste i efektywne. Pozwala na oszczędną gospodarkę pamięcią operacyjną, gdyż informacje o tych systemach są niezbedne jedynie w kilku miejscach programu Turbo Wiener™. Przyjęta metoda umożliwia pełną swobodę wyboru języka programowania do opisu systemu. Sam opis systemu jest kompilowany oddzielnie i nie jest konieczna powtórna kompilacja programu Turbo Wiener™.

Rozwiązanie równań różniczkowych opisujących system modelowany lub korygowany oraz wzorcowy, dokonywane jest w programie Turbo Wiener™ za pomocą metody Braytona-Gustavsona-Hatchela [6] (M. M. Stabrowski [51]).

Ponieważ model Laguerre'a zawierają wielomiany i funkcje Laguerre'a, to przy opracowywaniu programu Turbo Wiener™, koniecznym było określenie metody wyznaczania wartości wielomianów oraz funkcji Laguerre'a. Ze względu na dużą złożoność obliczeniową algorytmów identyfikacji i korekcji systemów, do wyznaczania wartości tych wielomianów zastosowano regułę trójczłonową, która zapewnia maksymalną szybkość obliczeń (J. Jankowska, M. Jankowski [30], s. 93). Z kolei przy obliczaniu splotów funkcji Laguerre'a z wymuszeniami zastosowano algorytm FFT, dla którego 256 wartości próbek funkcji Laguerre'a wyznaczane jest za pomocą kwadratury Gaussa-Legendre'a (patrz A. Ralston [41], s. 91).

Z przedstawionej wcześniej teorii wynika, że przy wyznaczaniu modelu systemu lub jego korektora, niezbędnym jest rozwiązanie układu równań liniowych. W omawianym programie zastosowano algorytm QR Francisa (J. Stoer, R. Bulirsch [52], s. 382 oraz G. Dahlquist, A. Björck [25], s. 197) z podwójnym ulepszeniem (G. Dahlquist, A. Björck [25], s. 200). Iloczyny skalarne występujące w pojawiających się tu układach równań normalnych zostały wyznaczone za pomocą złożonej kwadratury Newtona-Cotesa drugiego rzędu, nazywanej złożonym wzorem trapezów (S. Stoer, R. Bulisch [52], s. 128).

Jako źródło szumu białego wykorzystano w programie Turbo Wiener™ multiplikatywny generator liczb pseudolosowych zaproponowanych przez O. Tausski (D.

E. Knuth [32], s. 112), dla którego n-ta liczba pseudolosowa

$$r_n := 5^{15} r_{n-1} (\mathrm{mod} 2^{35}).$$

Generator ten wyróżnia się spośród innych generatorów liczb pseudolosowych, bardzo dobrymi parametrami widmowymi, związanymi z gęstością mocy (D. E. Knuth [32], s. 105).

5. PRZEBIEG OBLICZEŃ

Za pomocą programu Turbo Wienr™ wyznaczono *korektor Laguerre'a* dla *systemu inetrcyjnego pierwszego rzędu* opisanego za pomocą równania:

$$\tau \frac{dv}{dt} + v = ku,$$

gdzie k jest *współczynnikiem wzmocnienia* zdefiniowanym jako iloraz odpowiedzi v do wymuszenia u w stanie ustalonym, a τ jest *stałą czasową*, oraz *systemu oscylacyjnego* opisanego za pomocą równania:

$$T_0^2 \frac{d^2v}{dt^2} + 2\xi T_0 \frac{dv}{dt} + v = ku,$$

gdzie k jest *współczynnikiem wzmocnienia* zdefiniowanym tak jak poprzednio, T_0 jest *okresem drgań własnych nie tłumionych*, a $\xi \in [0,1]$ jest *względnym współczynnikiem tłumienia*. Parametry systemu intercyjnego pierwszego rzędu wynosiły tu:

$$k = 1 \text{ oraz } \tau = 10 \text{ s}.$$

Z kolei parametry systemu oscylacyjnego były tu:

$$k = 1,\ T_0 = 10 \text{ s oraz } \xi = 0.5.$$

Zbiory z opisem tych systemów mają nazwy inertlor.sys i oscilat.sys, a ich parametry zdefiniowane są w zbiorach inertlor.dat i oscilat.dat.

Za pomocą programu Turbo Wiener™ wyznaczono korektory dla systemu inercyjnego piewszego rzędu i systemu oscylacyjnego. Jako system wzorcowy przyjęto system inercyjny pierwszego rzędu o parametarach:

$$k = 1 \text{ oraz } \tau\text{-}1 \text{ s}.$$

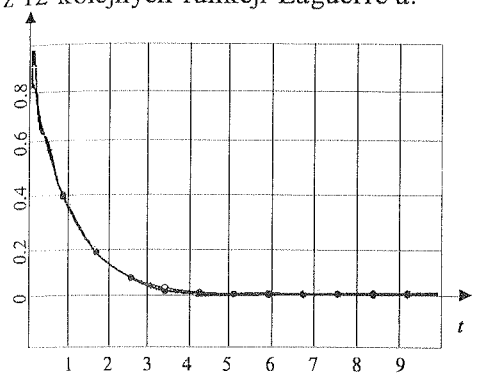
Zbiór z opisem tego systemu ma nazwę desired.sys, a jego parametry zdefiniowane są w zbiorze desired.dat. Utworzone zbiory z wynikami mają nazwy odpowiednio inertlor.cor i oscilat.cor.

Korektory systemu inercyjnego pierwszego rzędu i systemu oscylacyjnego wyznaczono jak dla modeli tych systemów, dla różnych wartości współczynnika widmowego. Współczynnik widmowy określa krotność próbek szumu białego i powi-

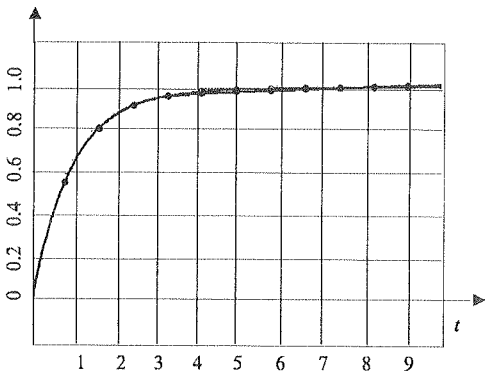
nien być potęgą liczby 2. Suboptymalna wartość współczynnika skali czasu

$$c = 5\frac{1}{2},$$

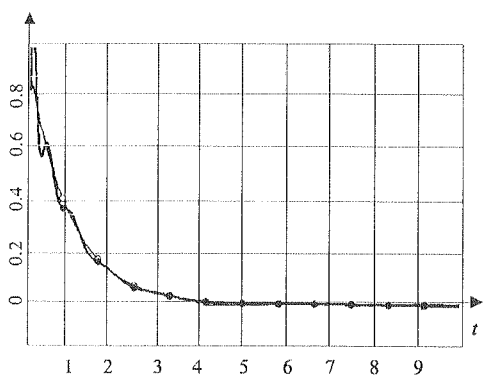
a współczynnika wudmowego wynosila 32. Ponadto wartość szumu była równa 1, a czas uśredniania wynosił 10 s. Korektor systemu inercyjnego składa się z 8 kolejnych funkcji Laguerre'a, a w przypadku systemu oscylacyjnego korektor składa się z 12 kolejnych funkcji Laguerre'a.



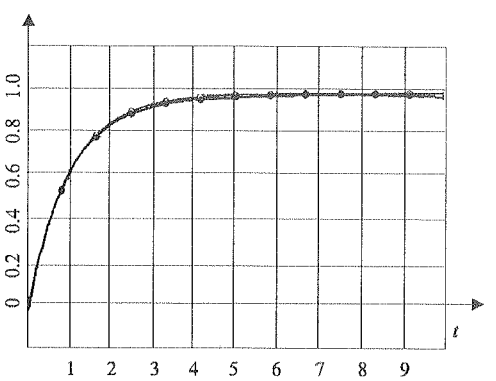
Rys. 5.1. Odpowiedź impulsowa wzorcowego systemu inercyjnego pierwszego rzędu ° oraz systemu inercyjnego po korekcji •.



Rys. 5.2. Odpowiedź skokowa wzorcowego systemu inercyjnego pierwszego rzędu ° oraz systemu inercyjnego po korekcji •.



Rys. 5.3. Odpowiedź impulsowa wzorcowego systemu inercyjnego pierwszego rzędu ° oraz systemu oscylacyjnego po korekcji •.



Rys. 5.4. Odpowiedź skokowa wzorcowego systemu inercyjnego pierwszego rzędu ° oraz systemu oscylacyjnego po korekcji •.

Otrzymane tu wyniki są bardz interesujące (rys. 5.1, 5.2, 5.3, i 5.4). W przypadku systemu inercyjnego (patrz rys. 8.1 i 8.2 z artykułu G. Ciesielskiego [23]) po korekcji uzyskano system, który zachowuje się jak system inercyjny i jest 10 razy szybszy pod względem błędu dynamicznego, od tegoż systemu inercyjnego przed korekcją (rys. 5.1

i 5.2). Najbardziej spektakularną jest jednak korekcja systemu oscylacyjnego (patrz rys. 8.3 i 8.4 z artykułu G. Ciesielskiego [23]). System ten po korekcji (rys. 5.3 i 5.4) zachowuje się jak system inercyjny i jest około 30 razy szybszy pod względem błędu dynamicznego, od tegoż systemu oscylacyjnego przez korekcją.

Wszystkie te wyniki są godne szczególnej uwagi.

6. PODSUMOWANIE

Za pomocą programu Turbo Wiener™ wyznaczono model Laguerre'a dla korektorów systemu inercyjnego i oscylacyjnego. Otrzymane wyniki są bardzo interesujące. Uzyskane bowiem wyniki bardzo dobrze opisują odpowiedzi impulsowe, jak i skokowe systemów inercyjnych i oscylacyjnych po korekcji. W przypadku korektora systemu inercyjnego stopień przybliżającego wielomianu Laguerre'a wynosił 7, a w przypadku systemu oscylacyjnego stopień ten wynosił 11. W przypadku systemu inercyjnego po korekcji uzyskano system, który zachowuje się jak system inercyjny i jest 10 razy szybszy pod względem błędu dynamicznego, od tegoż systemu inercyjnego przed korekcją. Najbardziej spektrakularną jest jednak korekcja systemu oscylacyjnego. System ten po korekcji zachowuje się jak system inercyjny i jest około 30 razy szybszy pod względem błędu dynamicznego, od tegoż systemu oscylacyjnego przed korekcją.

Autor artykułu pracuje obecnie nad nowym typem operatorów, nazywanych roboczo operatorami sklejanymi, które wykorzystują funkcje sklejane ([5], [7], [28] i [49]). Operatory te powinny odegrać podobną rolę, jaką odegrały kiedyś operatory Volterry i Wienera w teorii nieliniowych systemów stacjonarnych przyczynkowych ([39], [40] oraz [42]—48]).

BIBLIOGRAFIA

1. W. M. Amierbajew, N. A. Utimbajew: *Czislenyj analiz łagierrowskowo spiektra.* Nauka, Ałma-Ata, 1982.
2. J. S. Bendat, A. G. Piersol (tłum. a ang. J. Dudziewicz, R. Białobrzeski): *Metody analizy i pomiaru sygnałów losowych.* Biblioteka Naukowa Inżynieria, Warszawa, PWN, 1976.
3. P. Billingsley (tłum. z ang. K. Kizeweter, J. E. Rogulski): *Prawdopodobieństwo i miara.* Warszawa, PWN, 1987.
4. G. Birkhoff, T. C. Bartee (tłum. z ang. M. Szymańska): *Współczesna algebra stosowana.* Matematyka dla politechniki, Warszawa, PWN, 1987.
5. C. de Boor: *Practical to Splines.* Applied Mathematical Science, Vol. 27 New York, Springer-Verlag, 1978.
6. R. K. Brayton, F. G. Gustavson, G. D. Hatchel: *A new efficient algorithm for solving differential-algebraic systems using implicit backward differetiation formulas.* Proc. of IEEE, Vol. 60 (1), 1972 pp. 98—108.
7. G. Ciesielski: *Kształtowanie statystycznych charakterystyk czujników do pomiaru wielkości nieelektrycznych za pomocą przybliżeń funkcjami krzywikowymi.* Rozprawa doktorska, Instytut Podstaw Elektrotechniki, Politechnika Łódzka, Łódź, 1982.
8. G. Ciesielski: *Aproksymacja średniokwadratowa operatorów nieliniowych opisujących wielowymiarowe systemy pomiarowe.* XXIII Międzyuczelniana Konferencja Metrologów, Politechnika Warszawska, Warszawa, 1991 (referat nie opublikowany).

9. G. Ciesielski: *Modele Wienera wielowymiarowych systemów pomiarowych opisanych za pomocą operatorów nieliniowych. Modelowanie i symulacja systemów pomiarowych.* Materiały Sympozjum, AGH, Kraków, 1992, ss. 41 – 50.
10. G. Ciesielski: *Operatorowe modele wielowymiarowych systemów pomiarowych.* Seminarium Naukowe Komisji Kształcenia Komitetu Metrologii i Aparatury Naukowej Polskiej Akademii Nauk, 1994 (referat nie opublikowany).
11. G. Ciesielski: *Identyfikacja ogólnej struktury wielowymiarowych systemów pomiarowych. Modelowanie i symulacja systemów pomiarowych.* Materiały Sympozjum, AGH, Kraków, 1994, ss. 17 – 23.
12. G. Ciesielski: *Identyfikacja ogólnej struktury wielowymiarowych systemów stacjonarnych*, Kwartalnik Elektroniki i Telekomunikacji, T. 40, z. 3, 1994, ss. 419 – 428.
13. G. Ciesielski: *Modelowanie wielowymiarowych systemów stacjonarnych za pomocą splotów.* Kwartalnik Elektroniki i Telekomunikacji, T. 40, z. 3, 1994, ss. 429 – 448.
14. G. Ciesielski: *Modelowanie i korekcja wielowymiarowych stacjonarnych systemów pomiarowych za pomocą operatorów nieliniowych.* Rozprawa habilitacyjna, Politechnika Łódzka, Zeszyty Naukowe, Nr 699, Łódź, 1994.
15. G. Ciesielski: *Identyfikacja szczegółowej struktury wielowymiarowych systemów stacjonarnych.* Kwartalnik Elektroniki i Telekomunikacji, T. 40, z. 3, 1995, ss. 305 – 319.
16. G. Ciesielski: *Modelowanie wielowymiarowych systemów stacjonarnych przyczynowych.* Kwartalnik Elektroniki i Telekomunikacji, T. 41, z. 3, 1995, ss. 321 – 335.
17. G. Ciesielski: *Modelowanie wielowymiarowych liniowych systemów stacjonarnych w systemach pomiarowych. Modelowanie i symulacja systemów pomiarowych.* Materiały Sympozjum, AGH, Kraków, 1996, ss. 28 – 37.
18. G. Ciesielski: *Identyfikacja szczegółowej struktury wielowymiarowych systemów stacjonarnych w systemach pomiarowych. Modelowanie i symulacja systemów pomiarowych.* Materiały Sympozjum, AGH, Kraków, 1996, ss. 38 – 42.
19. G. Ciesielski: *Zastosowanie modeli Wienera w systemach pomiarowych. Systemy pomiarowe w badaniach naukowych i w przemyśle.* Materiały konferencyjne, Wyższa Szkoła Inżynierska, Zielona Góra, 1996, ss. 43 – 52.
20. G. Ciesielski: *Modelowanie wielowymiarowych systemów stacjonarnych w systemach pomiarowych, Modelowanie i symulacja systemów pomiarowych.* Materiały Sympozjum, AGH, Kraków, 1997, ss. 129 – 142.
21. G. Ciesielski: *Korekcja wielowymiarowych systemów stacjonarnych w systemach pomiarowych. Modelowanie i symulacja systemów pomiarowych.* Materiały Sympozjum, AGH, Kraków, 1997, ss. 143 – 150.
22. G. Ciesielski: *Korekcja wielowymiarowych systemów stacjonarnych przyczynowych.* Kwartalnik Elektroniki i Telekomunikacji (artykuł złożony do druku).
23. G. Ciesielski: *Korekcja liniowych systemów stacjonarnych za pomocą systemów Laguerre'a.* Kwartalnik Elektroniki i Telekomunikacji (artykuł złożony dso druku).
24. G. Ciesielski: *General structure of multidimensional time-invariant systems.* Communications on Pure and Applied Mathematics. John Wiley & Sons, Inc. (artykuł w przygotowaniu).
25. G. Dahlquist, Å. Björck (tłum. z ang. S. Paszkowski): *Metody numeryczne.* Warszawa, PWN, 1983.
26. P. Eykhof (tłum. z ang. A Bauer): *Identyfikacja w układach dynamicznych.* Biblioteka Naukowa Inżynieria, Warszawa, PWN, 1980.
27. R. J. P. de Figueiredo, T. A. W. Dwyer: *A best approximation framework and implementation for simulation of large-scale nonlinear systems* IEEE Trans. Circuits Syst., Vol. CAS-27, no 11, pp. 1005 – 1014
28. R. J. de Figueiredo: *Nonlinear circuits and systems applications of functional splines.* Proc. IEEE, 1982, pp. 1245 – 1247.
29. S. Gładysz: *Wstęp do topologii.* Matematyka dla politechniki, Warszawa, PWN, 1981.
30. J. Jankowska, M. Jankowski: *Przegląd metod i algorytmów numerycznych.* T. 1, Warszawa, WNT, 1981.

31. J. Jaworski, R. Morawski, J. Olędzki: *Wstęp do metrologii i techniki eksperymentu.* Warszawa, WNT. 1992.
32. D. E. Knuth: *The Art of Computer Programming.* Vol. 2, Massachusetts, Addison-Wesley, 1969.
33. J. Kudrewicz: *Częstotliwościowe metody w teorii nieliniowych układów dynamicznych.* Warszawa, WNT, 1970.
34. K. Kuratowski: *Wstęp do teorii mnogości i topologii.* Biblioteka Matematyczna, T. 9, Warszawa, PWN, 1980.
35. F. Morrison: *Sztuka modelowania układów dynamicznych, deterministycznych, chaotycznych, stochastycznych.* Warszawa, WNT. 1996.
36. J. Musielak: *Wstęp do analizy funkcjonalnej.* Warszawa, PWN, 1989.
37. J. Osiowski: *Zarys rachunku operatorowego. Teoria i zastosowanie w elektrotechnice.* Warszawa, WNT, 1981.
38. A. Papoulis: *Prawdopodobieństwo, zmienne losowe i procesy stochatyczne.* Warszawa, WNT, 1972.
39. J. Park, I. W. Sandberg: *Criteria for the approximation of nonlinear systems.* IEEE Trans. Circuits Syst.-I: Fundamental Theory and Applications, Vol. 39, No 8, 1992, pp. 673–676.
40. K. A. Pupkow, W. I. Kapalin, A. S. Juszczenko: *Funkcjnalnyje rjady w tjeorii nieliniejnych sistjem.* Moskwa, Nauka, 1976.
41. A. Ralston (tłum. z ang. R. Zuber): *Wstęp do analizy numerycznej.* Warszawa, PWN, 1983.
42. I. W. Sandberg: *Criteria for the respondense of nonlinear to be L-asymptotic periodic.* Bell Syst. Tech. J., Vol. 60, No 10, 1981, pp. 2359–2371.
43. I. W. Sandberg: *Series expansion for nonlinear systems.* Proc. IEEE, 1982, pp. 110–113.
44. I. W. Sandberg: *Expansion for nonlinear systems.* Bell Syst. Tech. J., Vol. 61, No 2, pp. 159–199.
45. I. W. Sandberg: *Volterra expansion for time-verying systems* Bell Syst. Tech. J., Vol. 61, No 2, 1982, pp. 201–225.
46. I. W. Sandberg: *Multidimensional nonlinear systems and structure theorems.* J. Circuits, syst. and Computers, Vol. 2, No 4, 1992, pp. 383–388.
47. I. W. Sandberg: *Approximately-finite memory and input-output maps.* IEEE Trans. Circuits syst.-I: Fundamental Theory and Applications, Vol. 39, No 7, 1992, pp. 549–556.
48. M. Schetzen: *The Volterra and Wiener Theories of Nonlinear Systems.* New York, John Wiley & Sons, 1980.
49. L. L. Schumaker: *Spline Functions. Basic Theory.* New York, John Wiley & Sons, 1981.
50. T. Söderström, P. Stoica: *Identyfikacja systemów.* Warszawa, PWN, 1997.
51. M. M. Stabrowski: *Pivoted block solvers for large banded linear equation systems.* communicat. In Num. Meth. In Eng., Vol. 3, 1997, pp. 407–415.
52. JH. Stoer, R. Bulirsch (tłum. z ang. J. Cytowski): *Wstęp na analizy numerycznej.* Warszawa, PWN, 1987.
53. J. Szabatin: *Podstawy teorii sygnałów.* Warszawa, WNT, 1982.
54. B. Widrow, S. D. Stearns: *Adaptive Signal Processing.* New Jersey, Prentice-Hall, Inc, 1985.
55. A. Wojnar: *Teoria sygnałów.* Podręczniki akademickie EIT, Warszawa, WNT 1980.
56. R. R. Yager, D. P. Filev: *Podstawy modelowania i sterowania rozmytego.* Warszawa, WNT, 1995.

G. CIESIELSKI

CORRECTION OF LINEAR AND TIME-INVARIANT SYSTEMS BY LAGUERRE SYSTEMS

Summary

The problem of structure and parameter identifications of correctors models of linear time-invariant system is presented in this paper. The structure of the Laguerre models is applied to they describes. Presented theory involes from uniform general operator's theory of systems linear as well as nonlinear. Uniform general operator's theory of systems has evaluated by author of this paper, form several years

([7] – [24]). In this paper, the Turbo Wiener™ program, designed and evaluated by autors for MS Windows operational system, is presented. This program lets to calculated the models and correctors of time-invariant systems, which are linear or nonlinear. There is applied uniform general operator's general theory of systems. This paper presents too the results of computation by use of program Turbo Wiener™ for correctors of typical linear time-invariant systems.

Key words: universal functional theory of systems, correction, modeling, structure identification, parametric identification, multidimensional systems, time-invariant systems, causal systems, operators, convolutions, stationary space, convolution space, causal space, program Turbo Wiener™.

Zast
pro
rozpozna

W
ciu o k
strumen
wym. C
z kame
cech op
twarzan
zamiesz
rozpozn

Słowa k
matyczn

Aby zro
cji, pozwal
należy zasta
pięciu kana
niu (wzrok
w każdej dz
pracy.

Zatem
które dzięk
- identyfik
- oceny ja

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2000, 46, z. 2, ss. 163–174

Zastosowanie systemów wizyjnych w automatyzacji procesów produkcyjnych na przykładzie systemu rozpoznającego detale podlegające operacjom montażowym

MIROSŁAW GAJER

Katedra Automatyki, Akademia Górniczo-Hutnicza w Krakowie

Otrzymano 2000.03.10
Autoryzowano 2000.04.17

W artykule zaproponowano wykorzystanie systemu wizyjnego, zrealizowanego w oparciu o kartę akwizycyjną z wieloprocesorowym układem TMS320C80 firmy Texas Instruments, do automatycznej identyfikacji obiektów podlegających operacjom montażowym. Opisano szczegółowo wszystkie kolejne operacje wykonywane na pozyskanych z kamery obrazach, takie jak wstępne przetworzenie obrazu, jego segmentację, ekstrakcję cech opisujących rozpoznawane obiekty oraz ich klasyfikację. Poszczególne etapy przetwarzania obrazów zostały zobrazowane na zamieszczonych w artykule ilustracjach oraz zamieszczono uzyskane dla prezentowanego przykładu wyniki procesu automatycznego rozpoznawania obiektów.

Słowa kluczowe: przetwarzanie obrazów, segmentacja obrazu, indeksacja obiektów, automatyczna klasyfikacja obiektów.

1. WSTĘP

Aby zrozumieć znaczenie systemów wizyjnych dla procesu automatyzacji produkcji, pozwalającego na eliminację osób pracujących na stanowisku montażowym, należy zastanowić się nad znaczeniem systemu wzrokowego u człowieka [1]. Spośród pięciu kanałów informacyjnych, dostarczających człowiekowi wiedzę o jego otoczeniu (wzrok, słuch, dotyk, smak i węch), wzrok posiada znaczenie dominujące w każdej dziedzinie aktywności człowieka, w tym także na każdym stanowisku jego pracy.

Zatem w systemy wizyjne wyposażone są coraz częściej roboty przemysłowe, które dzięki takiemu rozwiązaniu zyskują zdolność do [2]:

- identyfikacji elementów podlegających operacjom montażowym
- oceny jakości montowanych elementów

- określenia miejsca, w którym należy element uchwycić
- wyznaczanie punktu przestrzeni, w którym należy manipulowany element umieścić
- kontroli trajektorii ruch manipulowanych obiektów
- przechwycenia ruchomych obiektów
- nadzoru nad przestrzenią roboczą
- sterowania robotami mobilnymi

Ponadto do powszechnie wymienianych korzyści, jakie może przynieść wyposażenie robotów przemysłowych w systemy wizyjne można zaliczyć [3]:

- polepszenie jakości produkcji, w skutek kontroli montowanych elementów oraz wykonywanych operacji
- zwiększenie wydajności pracy i tempa produkcji
- zmniejszenie liczby odpadów i lepsze wykorzystanie materiałów zmniejszenie liczby wyrobów wadliwie zmontowanych
- eliminacja stanowisk ręcznej kontroli jakości wyrobów

Podsumowując powyższe rozważania należy podkreślić, że najważniejszym zadaniem realizowanym przez system wizyjny robota jest identyfikacja bądź weryfikacja montowanych elementów. Z reguły od wyniku przeprowadzonej identyfikacji zależą dalsze czynności wykonywane przez robota. Ponadto ocena i rozpznanie elementu może znaleźć zastosowanie w systemach, w których robot wspomaga proces sortowania lub pakowania elementów bądź został zastosowany na stanowisku automatycznej kontroli jakości produkcji [4].

2. BUDOWA I ZASADY FUNKCJONOWANIA SYSTEMU WIZYJNEGO

W artykule rozważono możliwość zastosowania systemu wizyjnego do identyfikacji obiektów, podlegających automatycznym operacjom montażowym. Założono, że rozpoznawanymi obiektami są podkładki do śrub o różnej średnicy, a celem systemu wizyjnego jest określenie ile takich obiektów znajduje się w polu widzenia kamery oraz podanie do jakich klas przynależności poszczególne podkładki należą. Dzięki wyposażeniu w system wizyjny, robot przemysłowy może w sposób samoczynny wybierać z zasobnika podkładki o żądanej średnicy. Rozwiązanie takie powoduje, że robot staje się bardziej elastyczny, zyskuje zdolność do adaptacji do zmiennych warunków środowiska pracy oraz zanika konieczność współpracy robota ze skomplikowanym systemem podajników, gwarantujących, że podkładka o odpowiedniej średnicy znajdzie się w danym momencie w z góry wyznaczonym miejscu.

Omawiany w artykule system wizyjny został zbudowany w oparciu o kamerę typu EVI G-21 (firmy Sony) dostarczającą obrazu w standardzie PAL oraz, współpracującą z komputerem PC, kartę akwizycji obrazu SDB (ang. Software Development Board) firmy Texas Instruments. Karta SDB, oprócz układu akwizycji obrazu (ang. frame grabber) została wyposażona we własną jednostkę przetwarzającą dane, w postaci wieloprocesorowego układu TMS320C80 firmy Texas Instruments. W skład układu TMS320C80 wchodzi 32-bitowy procesor nadrzędny typu RISC oraz 32-bitowe procesory DSP, specjalizowane do szybkiej realizacji operacji przetwarza-

nia obrazów. Firma Texas Instruments dostarczyła również narzędzia programistyczne w postaci dwóch kompilatorów języka C standardu ANSI (oddzielnych dla procesora RISC i procesorów DSP), linkera umożliwiającego połączenie skompilikowanych programów w jeden wykonywalny moduł oraz biblioteki funkcji, zarządzających pracą bloków akwizycji danych [5].

Automatyczna identyfikacja obiektów w systemie wizyjnym jest zawsze procesem wieloetapowym, składającym się z szeregu następujących kolejno po sobie czynności, wśród których można wyróżnić [6]:

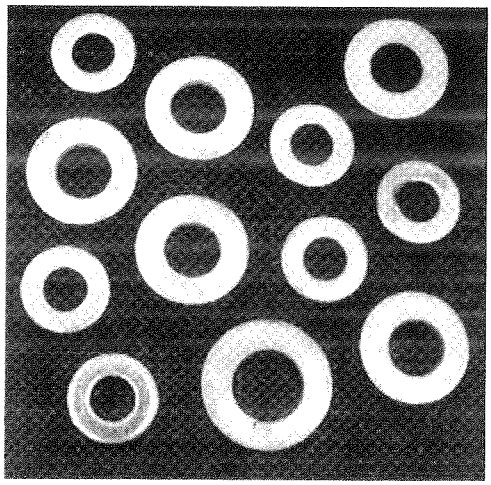
- Wstępne przetworzanie obrazu, mające na celu poprawę jego jakości, eliminacje ewentualnych zakłóceń bądź dodanego do sygnału obrazu w torze transmisyjnym szumu, a także zaakcentowanie i uwypuklenie w obrazie tych jego elementów, które posiadają istotne znaczenie z punktu widzenia dalszego procesu jego rozpoznawania.
- Segmentację obrazu, polegającą na wydzielaniu z obrazu obiektów podlegających identyfikacji. Rozważana operacja połączona jest zwykle z indeksacją rozpoznawanych obiektów, polegająca na zaznaczeniu każdego z nich odmiennym poziomem szarości.
- Pomiar wartości cech opisujących badane obiekty, którego celem jest uzyskanie wartości liczbowych, w oparciu, o które przprowadzone zostanie rozpoznawanie obiektów.
- Automatyczna klasyfikacja, polegająca na przypisaniu badanych obiektów do jednej ze zdefiniowanych uprzednio klas przynależności.

Wszystkie wymienione powyżej operacje przetwarzania i analizy obrazów zostały szczegółowo omówione w kolejnych punktach artykułu.

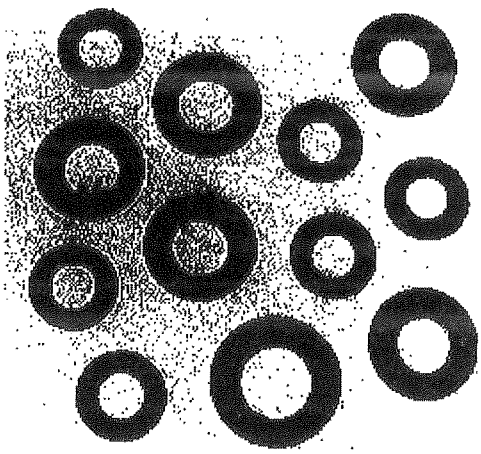
3. WSTĘPNE PRZETWARZANIE OBRAZÓW

Obrazy oglądanej przez system wizyjny sceny zostały pozyskane z kamery jako obrazy monochromatyczne o 256 dostępnych poziomach jasności pikseli oraz o rozdzielczości przestrzennej 512 na 512 pikseli. Zapewniono stały poziom oświetlenia analizowanej w systemie wizyjnym sceny, poprzez jej oświetlenie dodatkowym źródłem światła w postaci lampy halogenowej o mocy 20 W. Zapewniono także stałe oddalenie kamery od obserwowanej sceny, poprzez umieszczenie jej na nieruchomym statywie. Kamera została ponadto wyposażona w funkcję samoogniskowania obiektywu (ang. *autozoom*). Przykładowy obraz analizowanej sceny zawierającej nakrętki do śrub o różnej średnicy został pokazany na rys. 1.

Pokazany z kamery obraz został następnie poddany procesowi binaryzacji, polegającemu na przypisaniu każdemu z pikseli maksymalnego bądź minimalnego poziomu jasności (barwy białej bądź czarnej), w zależności od tego czy poziom jasności danego piksela znajdował się poniżej, czy też powyżej zadanej wartości progowej. Wartość progu dla procesu binaryzacji została ustalona eksperymentalnie na 60 poziomie jasności. Obraz po binaryzacji został zamieszczony na rys. 2.



Rys. 1. Obraz pozyskany z kamery.



Rys. 2. Obraz po binaryzacji.

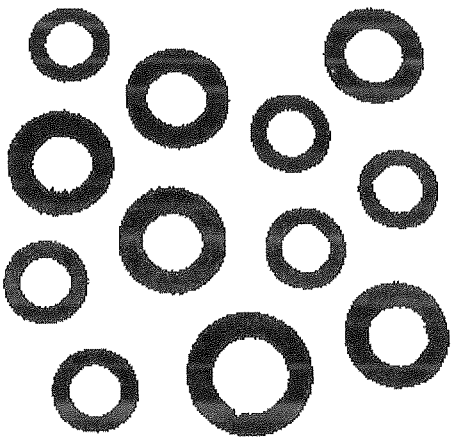
W kroku kolejnym obraz binarny został poddany filtracji, której celem była eliminacja zakłóceń występujących najczęściej w postaci pojedynczych czarnych pikseli usytuowanych na białym tle bądź też mających formę niewielkich zgrupowań takich pikseli.

W celu realizacji postawionego zadania, zastosowano operację erozji binarnej obrazu. Sposób działania rozważanej operacji polega na sprawdzeniu sąsiedztwa badanego piksela. Jeżeli badany piksel (x) posiada kolor czarny, wówczas po erozji binarnej piksel ten będzie w dalszym ciągu posiadał kolor czarny, tylko w przypadku, gdy wszystkie piksele sąsiednie (a, b, c, d) również posiadają kolor czarny. Postać maski wyznaczającej obszar sąsiedztwa piksela (x) została przedstawiona na rys. 3.

	a	
b	x	c
	d	

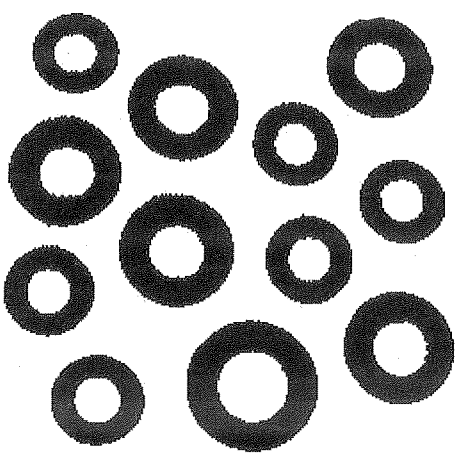
Rys. 3. Maska wyznaczająca obszar sąsiedztwa piksela.

Obraz po erozji binarnej został zamieszczony na rys. 4.



Rys. 4. Obraz po erozji binarnej.

Erozja binarna prowadzi do eliminacji zakłóceń w postaci drobnych zgrupowań czarnych pikseli rozrzuconych na białym tle, ale jednocześnie prowadzi do widocznego zwężenia i pomniejszenia analizowanych obiektów. W celu eliminacji powyższego mankamentu obraz poddany został dylatacji binarnej, która to operacja



Rys. 5. Obraz po dylatacji binarnej.

prowadzi do ponownego poszerzenia obiektów. Dla potrzeb dylatacji binarnej wykorzystywany jest obszar sąsiedztwa, zdeterminowany postacią maski przedstawionej na rys. 3. Podczas realizacji operacji dylatacji binarnej obrazu, badany biały piksel (x) uzyska kolor czarny, jeżeli co najmniej jeden z jego najbliższych sąsiadów (a, b, c, d) posiada również kolor czarny. Obraz po dylatacji został przedstawiony na rys. 5.

4. SEGMENTACJA OBRAZU

Celem segmentacji jest wydzielenie z obrazu poszczególnych obiektów oraz przeprowadzenie ich indeksacji, czyli zaznaczenie każdego z nich odrębnym poziomem szarości [4].

Segmentacja obrazu realizowana jest w ten sposób, że analizowane są kolejne piksele obrazu binarnego (wiersz po wierszu od górnej do dolnej krawędzi obrazu), aż do napotkania pierwszego czarnego piksela. Wówczas barwa takiego piksela zostaje zmieniona na wybrany aktualnie kolor indeksacji. W przypadku rozważanych obrazów, posiadających 256 poziomów szarości pikseli, do wyboru są 254 kolory indeksacji obiektów (wszystkie poziomy jasności oprócz białego i czarnego, gdyż te zostały już wykorzystane podczas binaryzacji obrazu). Następnie analizowane są dalsze piksele obrazu i jeżeli zostanie stwierdzone, że dany piksel posiada barwę czarną, wówczas pikselowi temu zostaje przypisany akualny kolor indeksacji, jedynie w przypadku, gdy co najmniej jedne z ośmiu najbliższych sąsiadów takiego piksela (a, b, c, d, e, f, g, h) został już zaindeksowany. Maska wyznaczająca obszar sąsiedztwa została przedstawiona na rys. 6.

a	b	c
d	x	e
f	g	h

Rys. 6. Maska wyznaczająca obszar sąsiedztwa w procesie indeksacji obiektów.

Rys. 7. Przykładowy obraz po indeksacji obiektów.

Realizacja opisanej powyżej procedury powoduje systematyczne dołączanie do zaindeksowanego obszaru kolejnych pikseli indeksowanego obiektu. Procedura ta powtarzana jest aż do momentu, gdy na obrazie nie występują już żadne zmiany, wówczas można uznać, że cały obszar badanego obiektu został już wypełniony zadanym poziomem szarości. Wtedy należy zmienić kolor indeksacji i rozpocząć poszukiwanie kolejnego obiektu. Przykładowy obraz po indeksacji przedstawionych na nim obiektów został przedstawiony na rys. 7. Przy czym kolejne poziomy indeksacji zwiększane były co dziesiąty stopień szarości pikseli.

5. KLASYFIKACJA OBIEKTÓW

Identyfikacja obiektów, polegająca na zaliczeniu ich do jednej z góry zdefiniowanych klas przynależności, została oparta na pomiarze pola powierzchni rozpoznawanych obiektów. Dla obiektów, wypełnionych kolejnymi poziomami indeksacji, dokonywany jest pomiar ich pola powierzchni poprzez zaliczenie reprezentujących jej pikseli. Następnie otrzymana w ten sposób liczba, jest porównywana z dwoma wartościami progowymi (górną i dolną), wyznaczonymi uprzednio dla każdej z uwzględnionych klas przynależności badanych obiektów. W ten sposób zaindeksowane obiekty zostają kolejno zaliczone do właściwych im klas przynależności.

Rezultaty procesu rozpoznawania przeprowadzonego dla zamieszczonych na rys. 7 obiektów zostały przedstawione w tabl. 1.

Tablica 1

Wyniki procesu rozpoznawania

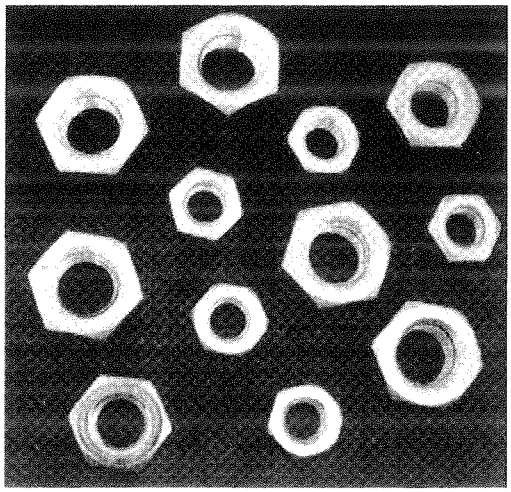
poziom	10	30	50	70	90	110	130	150	170	190	210	230
pole	4204	6427	7176	4283	7596	4278	7932	4560	4856	7451	9722	4922
klasa	1	2	2	1	2	1	2	1	1	2	3	1

W pierwszym wierszu tabl. 1 zamieszczono poziom jasności, który wykorzystano do indeksacji danego obiektu, przy czym kolejne poziomy indeksacji zwiększane były o 20 stopni jasności. Następnie w drugim wierszu tabeli zamieszczono, zmierzone w pikselach, pole powierzchni analizowanego obiektu. Z kolei w wierszu trzecim umieszczono numer klasy przynależności, do której system rozpoznający zaliczył badaną podkładkę.

6. PRZYKŁADY ROZPOZNAWANIA INNYCH OBIEKTÓW

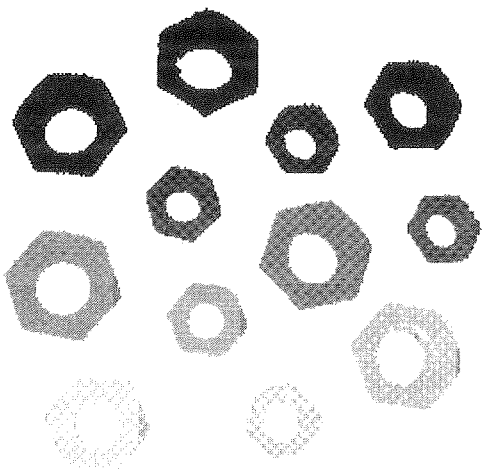
Analogiczne eksperymenty przeprowadzono także dla innych obiektów, które mogą podlegać zautomatyzowanym operacjom montażowym. Dla przykładu wybrano śruby, nakrętki oraz gumowe uszczelki.

Obraz pozyskany z kamery, przedstawiający rozpoznawane w systemie wizyjnym nakrętki do śrub został zamieszczony na rys. 8.



Rys. 8. Nakrętki rozpoznawane w systemie wizyjnym.

Obraz nakrętek po ich indeksacji został zamieszczony na rys. 9.



Rys. 9. Obraz nakrętek po indeksacji.

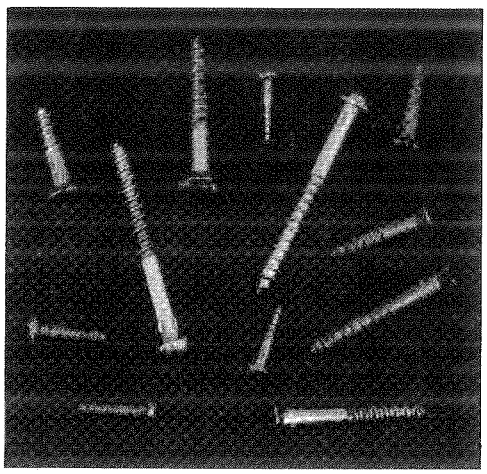
Wyniki rozpoznawania typów nakrętek do śrub zostały zamieszczone w tabl. 2.

Tablica 2

Wyniki procesu rozpoznawania

poziom	10	30	50	70	90	110	130	150	170	190	210	230
pole	6370	5383	6914	3168	3363	3163	7027	7250	3465	6875	5791	3460
klasa	1	2	1	3	3	3	1	1	3	1	2	3

Z kolei obraz przedstawiający rozpoznawane w systemie wizyjnym śruby został zamieszczony na rys. 10.



Rys. 10. Śruby rozpoznawane w systemie wizyjnym

Obraz śrub po ich indeksacji został zamieszczony na rys. 11.

Rys. 11. Obraz śrub po indeksacji.

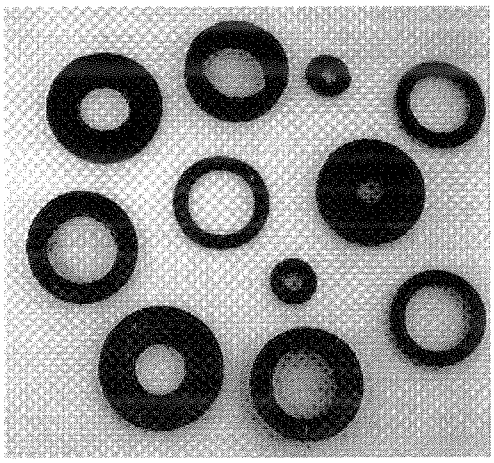
Wyniki rozpoznawania typów śrub zostały zamieszczone w tabl. 3.

Tablica 3

Wyniki procesu rozpoznawania

poziom	10	30	50	70	90	110	130	150	170	190	210	230
pole	2152	1000	663	3355	1262	3477	1186	1801	616	669	578	1877
klasa	3	2	1	4	2	4	2	3	1	1	1	3

Natomiast obraz przedstawiający rozpoznawane w systemie wizyjnym uszczelki gumowe został zamieszczony na rys. 12.



Rys. 12. Uszczelki rozpoznawane w systemie wizyjnym

Obraz uszczelek po ich indeksacji został zamieszczony na rys. 13.

Rys. 13. Obraz uszczelek po indeksacji.

Wyniki rozpoznawania typów uszczelek zostały zamieszczone w tabl. 4.

Tablica 4

Wyniki procesu rozpoznawania

poziom	10	30	50	70	90	110	130	150	170	190	210
pole	5877	8264	1265	3304	8567	3411	5995	1373	3837	10021	6039
klasa	3	4	2	1	4	1	3	2	1	5	3

Przeprowadzone eksperymenty rozpoznawania wykazały, że zaprezentowany w artykule system wizyjny może zostać skutecznie wykorzystany do rozpoznawnia szerokiej klasy typów detali podlegających operacjom montażowym na zautomatyzowanych liniach produkcyjnych, wykorzystujących roboty przemysłowe. Zastosowana metoda rozpoznawania pozwala na identyfikację obiektów posiadających podobny kształt, lecz różne pole powierzchni. W związku z powyższym nie może być ona zastosowana do identyfikacji obiektów o różnym kształcie i takim samym polu powierzchni, ponieważ obiekty takie są dla omawianego systemu wizyjnego identyczne. W takim przypadku należy wprowadzić wektor cech o większej liczbie wymiarów, niosących bardziej szczegółowe informacje o kształtach rozpoznawanych obiektów.

7. WNIOSKI KOŃCOWE

Poprawność działania zaproponowanego systemu wizyjnego została potwierdzona podczas eksperymentów rozpoznania różnej liczby obiektów (śrub, uszczelek, podkładek, nakrętek itp.), ustawionych w różnych konfiguracjach przestrzennych. Należy zaznaczyć, że wszystkie eksperymenty zakończyły się pełnym sukcesem, polegającym na zaliczeniu badanych obiektów do właściwej klasy przynależności. Uzyskanie tak dobrych rezultatów świadczy o tym, że pomiar pola powierzchni obiektów jest dobrym dyskryminatorem ich klas przynależności oraz może stanowić podstawę do praktycznych zastosowań przemysłowych.

Spośród opisanych w artykule operacji przetwarzania obrazów, przetwarzanie wstępne obrazu (binaryzacja, erozja binarna i dylatacja binarna) wymaga najdłuższego czasu pracy procesora, ponieważ rozważane operacje muszą zostać wykonane każdorazowo dla kilku setek tysięcy pikseli obrazu. Ponieważ, jednocześnie są to stosunkowo proste operacje oraz nie występują zbyt duże współzależności pomiędzy przetwarzanymi danymi, istnieje możliwość ich zrównoleglenia w łatwy i naturalny sposób [7]. Ponieważ zastosowany do prztwarzania obrazów system TMS320C80 został wyposażony w cztery równoległe procesory sygnałowe PP0, PP1, PP2 i PP3 (ang. Parellel Processors), przetwarzane obrazy należało podzielić na cztery równe części w postaci kwadratu o boku 256 pikseli i każdą z nich przydzielić do odbrębnego procesora PP. Praca utworzonej w ten sposób architektury SIMD (ang. Single Instruction Multiple Data) polegała na tym, że każdy z procesorów PP wykonywał te same operacje dla odrębnego zestawu danych. Jedynie w pewnych przypadkach (erozja i dylatacja binarna obrazu), wydzielone obszary musiały zachodzić na siebie na szerokość jednego piksela, co zostało zdeterminowane postacią maski przedstawionej na rys. 3.

Dzięki zaimplementowniu obliczeń równoległych czas realizacji operacji wstępnego przetwarzania obrazu wyniósł około 0,76 sekund, co stanowi prawie 3,17 krotne przyspieszenie obliczeń (ang. speedup), w porównaniu z ich realizacją na systemie jednoprocesorowym zrealizowanym w oparciu o procesor nadrzędny MP (ang. Master Processor) układu TMS320C80. Całkowity czas potrzebny na rozpoznanie zawartych w obrazie obiektów wynosił około 2,41 sekundy. Wynik taki uzyskano dla

dwunastu obiektów, trzeba bowiem mieć na uwadze, że czas potrzebny na wykonanie indeksacji obiektów zależy od ich liczby.

W literaturze przedmiotu można znaleźć opisy wielu różnych systemów wizyjnych wykorzystywanych do automatyzacji procesów montażowych. Dla przykładu w pracy [8] omówiono system wizyjny służący do lokalizacji obiektów poruszających się na taśmociągu. Z kolei w pracy [9] podano przykład systemu wizyjnego, którego celem było rozpoznawanie montowanych elementów bloków silników spalinowych.

BIBLIOGRAFIA

1. R. T a d e u s i e w i c z: *Problemy biocybernetyki.* Warszawa, PWN 1991.
2. P. T r e p a g n i e r: *Recent developments in visual gaging.* Conference Proceedings: Machnie Vision Association., Detroit, North Holland, 1985.
3. A. P u g h: *Robot Vision and Sensory Control.* Proceedings of the 4th International Conference Ro ViSec 4, KLondon, North Holland, 1984.
4. R. T a d e u s i e w i c z: *Systemy wizyjne robotów przemysłowych.* Warszawa, WNT, 1992.
5. *TMS320C80 System Level Synopis*; Materiały Katalogowe Firmy Texas Instruments, 1995.
6. H. Y i u: *Implementation of an image processing library for the TMS320C80*; Materiały Katalogowe Firmy Texas Instruments, 1997.
7. B. K w o l e k: *Komputerowe przetwarzanie obrazów w ieloprocesorowym systemie czasu rzeczywistego.* Materiały konferencyjne, VI Konferencja Systemów Czasu Rzeczywistego, Zakopane '99, 27–30 wrzesień 1999, ss. 271–280.
8. J. A m a t, A. C a s a l s, V. L l a r i o: *Location of work-pieces and guidance of industrial robots with a vision systems.* Robot Vision and Sensory Controls. Proceedings of the 4th International Conference Ro ViSec, London, 1984, pp. 223–230.
9. H. L i n d e r m a n. H. Hollek: *Supporting sensors for vision guided robots.* Proceedings of the 5th International Conference on Robot Vision and Sensor Controls. North Holland, Amsterdam, 1985.

Pracę zrealizowano w ramach grantu KBN 8T11C 03914 „Analiza i projektowanie systemów czasu rzeczywistego o różnym stopniu rozproszenia"

M. GEJER

THE USAGE OF VISION SYSTEMS FOR THE AUTOMATION OF ASSEMBLY PROCESSES

Summary

In the paper the usage of the vision system, basing on the acquisition board with the TMS320C80 Texas Instruments multiprocessor system, for automatic montage elements recognition is proposed. All image processing operations performed on the acquired from the camera imege are the oroughly described. Particularly the image pre-processing operations, image segmentation, feature extraction and classification are described in detalis. Consequent image processing stages are illustrated in figures placed in the paper and some example object cleassification results are also presented.

Key words: image processing, image segmentation, feature extraction, automatic image recognition

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2000, 46, z. 2, ss. 175–193

Comparison of three methods for neurofuzzy model initialization and their application to circuit performance modeling

RYSZARD HORBOWSKI

Wydział Elektroniki, Informatyki i Telekomunikacji,
Politechnika Gdańska

Received 1999.12.16
Authorized 2000.03.30

In the paper three methods of neurofuzzy model initialization are presented and experimentally verified. The first of these methods, which is typical for neural network, is the random initialization. As the second method, so-called pseudo-random initialization is proposed. This method in a simple way makes use of the available training set: the randomly chosen training points initialized model parameters responsible for the distribution of the membership function which define the linguistic labels of fuzzy rules. Such an approach is possible bacause of the physical interpretation of the model parameters.

The presentd experimental results show that average results obtained using the quasin-random parameter initialization overperform corresponding results obtained using the random initialization. Moreover, the quasi-random initialization can reduce or even eliminate an overfitting to the training data, which can appear for models having big number of parameters.

As the third method the mountain clustering-based initialization is considered. The experiments show that this method can give better results than the mentioned random and quasi-random method. However, the results of this method heavily depend on the values of two parameters, which control the process of the clustering. Unfortunately, the performed expreiments have not provided the optimal values of these parameters.

Additionally, the paper focuses on some aspects and problems which are connected with the presented modeling technique. The improvement of the quasi-random method is suggested, as well.

Key words: fuzzy- and neurofuzzy modeling, parameter estimation and initialization, adaptive modeling, circuit performance modeling.

1. INTRODUCTION

The neurofuzzy modeling, based on the developed in the 70th fuzzy reasoning, has been shown to be an interesting paradigm of nonlinear, large scale system modeling. Its applications range from a cancer diagnosis [1], on the one hand, to a chaotic time

series prediction [2], [3] and dynamics of autonomous underwater vehicle modeling [4], on the other hand. Recently the neurofuzzy modeling has been successfully applied to the circuit performance modeling [5], [6], as well.

In this paper we touch upon the class of neurofuzzy models which were used in the work [5]. Starting from IF-THEN fuzzy rules it is possible to express analytically the output of the model, which makes it possible to train the model parameters based on the training set consisted of input-output data paris. The transition from the fuzzy rules to the analytical expression of the model output is known as the Simplified Method of Fuzzy Reasoning (SMFR), and can be found in literature [4], [7], [8]. Some authors (e.g. [2], [5]) use explicit structures which are based on the SMFR without presenting a transition from a linguistic to the neurofuzzy model.

Properties (i.e. accuracy and generalization abilities of the neurofuzzy models with fixed structure, as for each parametrized model, depend on the set of the model parameters. These parameters are commonly identified based on the gradient descent algorithm: in each iteration of the algorithm the model parameters are updated toward minimization of the error measure. If the model output is linearly dependent on the model parameters, the gradient descent algorithm is always able to find a global minimum of the model error in the model parameter space. In the adverse situation, if the model output depends nonlinear on the model parameters, the gradient descent algorithm finds one of the local minima of the model error. In this case final parameter values, the speed of the training convergence, the final value of the error measure for the training and checking set and the generalization abilities of the model depend strongly on the initial values of the model parameters.

One of the main advantages of the neurofuzzy systems is their transparency. This means that available knowledge about the modeled system or process can be incorporated into the model construction and parameter initialization. Unfortunately, such an approach to the parameter initialization cannot be applied to the automatic model construction, and therefore we have to seek for other, automatic model initializers.

Probably the most frequently used initialization method for the neural networks is the random initialization. Because the neurofuzzy models being considered in this paper belong to the wide class of neural networks [8], [9], the random initialization technique can be applied to them, and this method was used (giving quite good results) in the mentioned work [5]. However, the random initialization do not make use of any additional information. As an improvement of the random initialization we propose a simple method which makes use of the data contained in the training set: randomly chosen training points fix initial values of the model parameters.

The third method of the parameter initialization we consider in this paper is based on the mountain clustering technique. The heuristic mountain clustring makes use of each available training data in order to find the fixed number of clusters. Thease clusters are subsets of the input-output space, in which density of the data points is large. Each cluster initializes the parameters of one fuzzy rule.

The paper is arranged as follows. In Chapter 2 basic assumptions, the model structure and the model training are presented. Three mathods of the parameter initialization are presented in Chapter 3. Chapter 4 describes the obtained experimental results whereas in Charper 5 these results are discussed.

2. BASIC ASSUMPTIONS. MODEL STRUCTURE AND TRAINING

Let $f(\mathbf{x})$ ba an unknown nonlinear function of interest. The vector $\mathbf{x}=(x_1, x_2,\ldots,x_N)$ consists of N elements (input variables) and the i-th element (the i-th input variable) x_i belongs to the known closed interval $X_i=[x_i^L; x_i^U]$. The input space of interest, $\mathbf{X}$, is defined as a Cartesian product:

$$\mathbf{X}=X_1\times X_2\times\ldots\times X_N. \tag{1}$$

Let us assume further that values of $f(\mathbf{x})$ belong to the closed interval $Y=[y^L; y^U]$. We name this interval the output space. Note however, that bacause $f(\mathbf{x})$ is usually unknown, the output space is unknown, as well.

Ler a linguistic model [7], [9], [11] be described by means of K fuzzy rules of the form:

$$\mathrm{r}_j\text{:IF } \bar{x}_1\text{IS } \mathrm{A}_{1j} \text{ AND } \bar{x}_2\text{IS A}_{2j} \text{ AND... AND } \bar{x}_N\text{IS A}_{Nj} \text{ THEN } \bar{y}\text{IS B}_j \tag{2}$$

Here A_{ij} denotes a linguistic label associated with the i-th linguistic variable $\bar{x}_i$ (being associated with the i-th input variable x_i) and with an antecedent of the j-th fuzzy rule, B_j denotes a linguistic label associated with a consequent of the j-th rule and $\bar{y}$ denotes a fuzzy output of the model.

Based on the fuzzy reasoning the linguistic model described by means of the fuzzy rules of the form (2) is able to infer a fuzzy output value of the model having actual values (appropriate fuzzy sets) of the linguistic variables $\bar{x}_i$. However, in order to enable the linguistic model to interact with the crisp (non-fuzzy) values, suitable fuzzi- and defuzzification producedures have to be performed. Their goal is to transform the crisp input values, x_i, into the fuzzy values of the linguistic variables $\bar{x}_i$ (fuzzification) and to transform the fuzzy output value, $\bar{y}$, into crisp value (defuzzification). This is illustrated in Fig. 1.

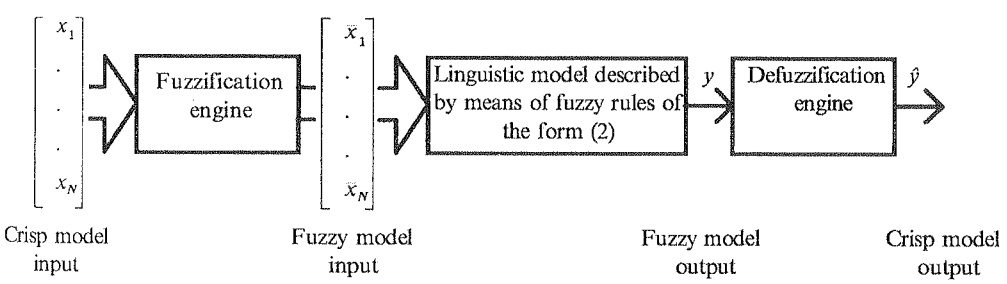


Fig. 1. Schematic representation of the multiple input – single output fuzzy model. Fuzzification and defuzzification engines enable the linguistic model to operate on crisp values

It can been shown that making some assumptions related to the fuzzy reasoning and to the fuzzi- and defuzzification procedures, the crisp output of the model of Fig. 1 can be expressed as follows:

$$\hat{y}(\mathbf{x}) = \frac{\sum_{j=1}^{K} \tau_j(\mathbf{x}) y^*_j}{\sum_{j=1}^{K} \tau_j(\mathbf{x})} \tag{3}$$

Here τ_j is a firing level of the j-th rule and y^*_j is a center of area (which is defined as $y^*_j = \int_Y y \mu_{B_j}(y) dy \Big/ \int_Y \mu_{B_j}(y) dy$) of the symmetrical membership function, $\mu_{B_j}(y)$, which defines the linguistic label B_j associated with the antecedent of the j-th rule. Equation (3) is named the simplified method of fuzzy reasoning; its derivation and assumptions it is based on can be found in the literature [7], [9], [11].

Assuming that the fuzzy AND operator is based on the product T-norm , the firing level of the j-th rule can be expressed as:

$$\tau_j(\mathbf{x}) = \prod_{i=1}^{N} \mu_{ij}(x_i) \tag{4}$$

where $\mu_{ij}(x_i)$ is a membership function defining the i-th linguistic label A_{ij} associated with the j-th fuzzy rule. Other definitions of the firing level are possible, but they lead to the more complicated training rules.

If we assume the linguistic label A_{ij} to be represented by the gaussian-shaped membership function:

$$\mu_{ij}(x_i) = \exp\left(-\frac{1}{2} \left(\frac{x_i - x^*_{ij}}{\sigma^*_{ij}} \right)^2 \right) \tag{5}$$

then the formula (3) can be rewritten as

$$\hat{y}(x) = \frac{\sum_{j=1}^{K} \prod_{i=1}^{N} \mu_{ij}(x_i) \tau_j}{\sum_{j=1}^{K} \prod_{i=1}^{N} \mu_{ij}(x_i)} = \frac{\sum_{j=1}^{K} \exp\left(-\frac{1}{2} \sum_{i=1}^{N} \left(\frac{x_i - x^*_{ij}}{\sigma^*_{ij}} \right)^2 \right) y^*_j}{\sum_{j=1}^{K} \exp\left(-\frac{1}{2} \sum_{i=1}^{N} \left(\frac{x_i - x^*_{ij}}{\sigma^*_{ij}} \right)^2 \right)} \tag{6}$$

Here x^*_{ij} and σ^*_{ij} denote a center and a width factor of the membership function defining the linguistic label A_{ij}, respectively. Consequently, each model parameter, i.e. y^*_j, x^*_{ij} and σ^*_{ij} has its physical interpretation, which distinguisches the investigated model from the other neural network-based models. This fact enable us to make use of the available knowledge about the modeled function during the parameter initialization.

Let $P = \{(\tilde{\mathbf{x}}_i, \tilde{y}_i)\}_{i=1}^{P}$ be the training set consisted of P data pairs of the form $(\tilde{\mathbf{x}}_i, \tilde{y}_i)$, where $\tilde{\mathbf{x}}_i \in \mathbf{X}$ and $\tilde{y}_i = f(\tilde{\mathbf{x}}_i)$. In other words, the training set consists of samples of the invetigated function f taken for P vectors which belong to the input space $\mathbf{X}$.

Let

$$p_i = \frac{1}{2} (\hat{y}(\tilde{\mathbf{x}}_i) - \tilde{y}_i)^2 \tag{7}$$

be en error measure of the model for the i-th element of P. If for the l-th element of the training set $(\tilde{\mathbf{x}}_l, \tilde{y}_l)$, the model parameters are iteratively updated according to the formulas:

$$x_{ij}^*(t+1) = x_{ij}^*(t) - \eta_{x_{ij}}(t)\frac{\partial p_l(t)}{\partial x_{ij}^*}$$

$$\sigma_{ij}^*(t+1) = \sigma_{ij}^*(t) - \eta_{\sigma_{ij}}(t)\frac{\partial p_l(t)}{\partial \sigma_{ij}^*} \tag{8}$$

$$y_j^*(t+1) = y_j^*(t) - \eta_{y_j}(t)\frac{\delta p_l(t)}{\partial y_j^*}$$

then the error measure (7) reaches, in general, its local minimum in the parameter space. The formulas (8) are known as the gradient descent training rules.

A single update of the model parameters according to (8) we name an iteration, whereas the set of such iterations performed only once for each training data we call an epoch. It can be shown that after each epoch the mean squared error (MSE):

$$\mathrm{MSE} = \frac{2}{P}\sum_{i=1}^{P} p_i = \frac{1}{P}\sum_{i=1}^{P}(\hat{y}(\tilde{\mathbf{x}}_i) - \tilde{y}_i)^2 \tag{9}$$

decreases and if the epochs are repeated, the MSE value reaches its local minimum.

For the investigated model (equations [3] – [6]) the derivatives appearing in (8) can be analytically obtained [7], [9], which significantly speeds-up the training process. The parameters $\eta_{x_{ij}}(t)$, $\eta_{\sigma_{ij}}(t)$ and $\eta_{y_j}(t)$ influence the speed and the stability of the training process. In general, the values of these parameters depend on the values of the modeled function and on the values of the input variables. In order to unify the approach of setting the values of these parameters we asume that the values of the input variables and the values of the modeled function have the same orders of magnitude. This assumption enable to set these parameters to have the same values. Because, in practice, the values of the input variables and the values of the modeled functions have different order of magnitude, they have to be either normalized or presceled. Additionally, the presented further mountain clustering requires the input variable values and the modeled function values to be normalized. Therefore, in the remainder of the paper, we assume that the input variable ranges, X_i, and the modeled function values are normalized to unity intervals [0,1]. This normalization can be performed by means of the linear transformation:

$$x_i' = \frac{x_i - x_i^L}{x_i^U - x_i^L},\quad y' = \frac{y - x^L}{y^U - y^L} \tag{10}$$

where x_i and y are the values of the input variables and modeled function before normalization and x_i' and y' after normalization, respectively. Based on the normalized values one can easily obtain the actual values:

$$x_i = x_i'(x_i^U - x_i^L) + x_i^L,\quad y = y'(y^U - y^L) + y^L \tag{11}$$

In the remainder of the paper we assume that we deal with normalized values. As a result of this we obtain: $\mathbf{X}=[0,1]^N$ and $\mathrm{Y}=[0,1]$.

3. THREE METHODS OF PARAMETER INITIALIZATION

The parameter initialization relies on fixing the initial values of the model parameters: $y_j^*(0)$, $x_{ij}^*(0)$ and $\sigma_{ij}^*(0)$. Below three mentioned methods are presented: the random initialization, the quasi-random initialization and the mountain clustering initialization.

3.1. RANDOM PARAMETER INITIALIZATION

This is a very simple and quick method: provided the input and output spaces are known, the model parameters are initialized according to the rules:

$$\begin{aligned} x_{ij}^*(0) &= rand(\mathrm{X}_i) \\ y_j^*(0) &\text{-} rand(\mathrm{Y}) \end{aligned} \tag{12}$$

where $rand([a; b])$ samples randomly the interval $[a; b]$ with the uniform probability. Because, as it was mentioned, the output space of the model, Y, is unknow, we have to estimate it. This estimate can be easily achieved by looking up the training set:

$$\begin{aligned} y^L &\approx \min_{i=1,\ldots,P} \{\tilde{y}_i\} \\ y^U &\approx \max_{i=1,\ldots,P} \{\tilde{y}_i\} \end{aligned} \tag{13}$$

The better the training set represents the modeled function, the better approximation can be achieved.

3.2. QUASI-RANDOM PARAMETER INITIALIZATION

This method can be regarded as a simple improvement of the method presented previously. Here, the initial values of the model parameters are obtained according to the rules:

$$\begin{aligned} x_{ij}^*(0) &= \tilde{x}_{k_j,i} \\ y_j^*(0) &= \tilde{y}_{k_j} \end{aligned} \tag{14}$$

where $k_j = rand\{1,\ldots,P\}$ for $j=1,\ldots,K$ is a randomly (without repetitions) chosen number from the set $\{1,\ldots,P\}$ ($\tilde{x}_{ij}$ denotes i-th element of the vector $\tilde{\mathbf{x}}_j$). In other words, we choose randomly K training data and use them to initialize the prameters. The advantages of this method are the same as of the previous method, i.e. the method is simple and quick.

3.3. MOUNTAIN CLUSTERING-BASED PARAMETER INITIALIZATION

Mountain clustering [7] is a heuristic method which clusters the training points in the input-outpus spece. Clusters are subsets of the input-output space, in which the density of the training points is the highest and which contain similar training points. The mountain clustering consists of three fundamental steps:

- a mountain function construction and
- a stepwies mountain function degradation, during which the centers of the clusters are localized.

In order to construct the mountain function we must have a set of basis points on which the mountain function is constructed. Assume $\mathrm{B}=\{(\hat{\mathbf{x}}_i, \hat{y}_i)\}_{i=1}^{B}$, $(\hat{\mathbf{x}}_i, \hat{y}_i)\in \mathrm{X}\times \mathrm{Y}$ is a set of B basis points. To each basis point we assign the value of the mountain function, O, according to the formula:

$$O(\hat{\mathbf{x}}_k, \hat{y}_k)=\sum_{i=1}^{P}\exp(-\beta \,\|\, (\hat{\mathbf{x}}_k, \hat{y}_k), (\tilde{\mathbf{x}}_i, \tilde{y}_i) \,\|\,) \tag{15}$$

where $\|\ \|$ denotes a distance measure between basis and training points, and β is a constant parameter.

After the mountain function evaluation, we find the first center $(\hat{\mathbf{x}}^{(1)}, \hat{y}^{(1)})$ of the first cluster:

$$(\hat{\mathbf{x}}^{(1)}, \hat{y}^{(1)})=(\hat{\mathbf{x}}_k, \hat{y}_k):O(\hat{\mathbf{x}}_k, \hat{y}_k)=\max_{i=1,\ldots,B}\{O(\hat{\mathbf{x}}_i, \hat{y}_i)\} \tag{16}$$

Next, we perform the degradation of the mountain function according to the formula:

$$O'(\hat{\mathbf{x}}_i, \hat{y}_i)=O(\hat{\mathbf{x}}_i, \hat{y}_i)-O(\hat{\mathbf{x}}^{(1)}, \hat{y}^{(1)})\cdot\exp\left(-\gamma \,\|\, (\hat{\mathbf{x}}_i, \hat{y}_i), (\hat{\mathbf{x}}^{(1)}, \hat{y}^{(1)}) \,\|\,\right) \tag{17}$$

where γ is a constant parameter.

For the modified function we find the center of the second cluster, $(\hat{\mathbf{x}}^{(2)}, \hat{y}^{(2)})$, and repeat the degradation K-2 times to obtain K centers: $(\hat{\mathbf{x}}^{(j)}, \hat{y}^{(j)})$, $j=1,\ldots,K$. Each center initialize the model parameters:

$$\begin{aligned} x_{ij}^{*}(0)&=\hat{x}_i^{(j)} \\ y_j^{*}(0)&=\hat{y}^{(j)} \end{aligned} \tag{18}$$

for $i=1,\ldots,N$ and $j=1,\ldots,K$.

The main problem associated with the presented method realies on fixing the values of two parameters: β and γ. The first of them controls an effective size of the clusters whereas the second one controls the degradation process. Based on several experiments we tried to determine optimal values of these parameters for the normalized input-output space. Unfortunately, we have not achieved this. We have only established ranges of these parameters, for which good results can (but do not have to) be obtained; for the normalized input and output spaces these ranges are similary and equal β, $\gamma\in[2;8]$.

For the distance measure we used the Euclidean norm:

$$\| (\hat{x}_j, \hat{y}_j), (\tilde{x}_k, \tilde{y}_k) \| = \sum_{i=1}^{N} (\hat{x}_{ij} - \tilde{x}_{ki})^2 + (\hat{y}_j - \tilde{y}_k)^2 \tag{19}$$

Taking into account the assumed distance measure it becomes clear that in order to ensure an equal influence of the individual components in (19) on the mountain function values, the intervals X_i and Y must be identical.

The presented three methods initialize parameters which are responsible for the location of the membership functions. However, we did not mention how to initialize parameters σ_{ij}^* responsible for the widths of the membership functions. This problem arises from the fact that neither lower nor upper limitations of these parameters are known. One of the possible solutions to this problem is to assume the initial values of these parameters based on the previously performed experiments, and this approach is used in the remainder of this paper. The normalization of the intervals X_i and Y enable to pick up equal values of these parameters and to compare results obtained for different values of these parameters.

4. EXPERIMENTAL RESULTS

In this chapter three examples of the neurofuzzy model initialization are presented. Within each example the average MSE values, calculated for the training set and for the independently generated checking set, are presented. The averages are calculated after ten performances of the random and quasi-random rule initialization for the models with the fixed number of the rules. Additionally, for each mean value a normalized value of the variance, S_{10}^2, is supported; it is calculated according to:

$$S_{10}^2 = \frac{\sum_{i=1}^{10} (\mathrm{MSE}_i - \overline{\mathrm{MSE}})^2}{10 \cdot \mathrm{MSE}^2} \tag{20}$$

The values of the variance indicate only repeatability of the numerical results; neither the final model structures nor the interpretability of the final fuzzy rules were taken into account. At this moment it is worth to mention that initial linguistic labels (if used) during the training process can lose their initial meaning [10]. This is why the parameter initialization should ensure not only the small final error value of the model, but repeatability of the final model structures, as well. Nevertheless, the author of the paper is of the opinion that the interpretability of the investigated non-lattice neurofuzzy models is rather poor. Because of this such models, despite being based on the fuzzy reasoning, are usually used as block-box models (e.g. [5]) and this is why in the remainder of this paper we consider these models only from the numerical point of view.

Apart from the results obtained for the random and quasi-random initializations the results obtained based on the mountain clustering initialization are partly presented. Namely, for each example we performed mountain-clustering based

initialization for the following sets of (β, γ) pairs (see Section 3.3): (2.0,2.0), (2.0,4.0), (2.0,6.0), (2.0,8.0), (4.0,2.0), (4.0,4.0), (4.0,6.0), (4.0,8.0), (6.0,2.0), (6.0,4.0), (6.0,6.0), (6.0,8.0), (8.0,2.0), (8.0,4.0), (8.0,6.0), (8.0,8.0). Below only the best results and the corresponding values of the parameters β and γ are presented.

For all experiments we assumed $\sigma_{ij}^*(0)=\sigma^*(0)$ and $\eta_{x_{ij}}(t)=\eta_{\sigma_{ij}}(t)=\eta_{y_j}(t)=\eta=0.1$ for $i=1,\ldots,N$ and $j=1,\ldots,K$. The experiments was executed for two initial values of the parameter $\sigma^*(0)$: 0.1 and 0.4. Each initialized model was trained within 1000 epochs of the gradient descent algorithm.

Example I. A three-variable nonlinear function

As the first examle we have chosen a simple, nonlinear, tree-variable function of the from:

$$f(\mathbf{x})=kx_1^2e^{x_2\sin(\pi x_3)} \tag{21}$$

where $x_i\in X_i=[0,1]$ for $i=1,2,3$, and $k=1/e$ is a normalization coefficient which ensures that $Y=[0,1]$.

For the investigated function 216 training data and 125 checking data were sampled uniformly from the input range $[0.0,1.0]\times[0.0,1.0]\times[0.0,1.0]$ and $[0.1,0.9]\times[0.1,0.9]\times[0.1,0.9]$, respectively. Tables 1 – 6 present the obtained results.

The results presented in Table 1 and Table 2 allow to compare two different methods of the initialization for the fixed initial value of the parameter $\sigma^*(0)=0.4$, whereas Tables 3 and 4 enable to do the same comparison for $\sigma^*(0)=0.1$. Moreover, taking into account Table 1 and Table 3 or Table 2 and Table 4 we can compare

Table 1

The average MSE values and their variances for the three-variable function modeling; random initialization, $\sigma^*(0) = 0.4$

Number of rules	MSE/variance for the training set	MSE/variance for the checking set
4	8.11e-4/1.19e0	7.10e-4/7.91e-1
8	4.46e-5/2.39e-1	1.21e-3/2.26e0
16	1.29e-5/2.41e-1	1.61e-3/2.90e0

Table 2

The average MSE values and their variances for the three-variable function modeling; quasi-random initialization, $\sigma^*(0) = 0.4$

Number of rules	MSE/variance for the training set	MSE/variance for the checking set
4	1.57e-4/1.82e-1	2.22e-4/1.42e0
8	1.91e-5/1.33e-1	2.52e-5/3.47e-1
16	9.89e-6/3.63e-1	1.56e-5/1.02e-1

Table 3

The average MSE values and their variances for the three-variable function modeling; random initialization, $\sigma^*(0) = 0.1$

Number of rules	MSE/variance for the training set	MSE/variance for the checking set
4	5.65e-3/3.92e0	4.87e-3/5.72e0
8	4.43e-3/6.87e0	4.84e-3/2.64e0
16	1.04e-4/1.81e0	4.44e-3/2.57e0

Table 4

The average MSE values and their variances for the three-variable function modeling; quasi-random rule initialization, $\sigma^*(0) = 0.1$

Number of rules	MSE/variance for the training set	MSE/variance for the checking set
4	1.69e-3/2.81e-1	8.68e-4/2.06e-1
8	2.43e-4/2.07e0	2.85e-4/1.22e0
16	2.41e-5/5.74e-1	3.50e-4/3.25e0

results obtained for two different initial values of the parameter σ^* for the same initialization technique.

Let us take into consideration Tables 1 and 2. The results show the superiority of the quasi-random rule initialization over the random initialization. This becomes clear if we consider the MSE values calculated for the checking set which indicate the true quality of the model: the MSE values for this set are much better for the models initialized in a quasi-random fashion in comparision with the models initialized in a random fashion. Moreover, the repeatability of the results for the quasi-random initialization is generally better than for the random initialization. Tables 3 and 4 confirm the previously driven conclusions for other initial value of the parameter σ^*. Note however that the results presented in Tables 3 and Table 4 ($\sigma^*(0)=0.1$) are worse than the corresponding results obtained for $\sigma^*(0)=0.4$. This fact is discussed later.

In Table 5 and Table 6 the best results obtrained using the mountain clustering initialization are presented. The basis set was consisted of 625 basis points sampled uniformly from the input-output range $[0.1,0.9]\times[0.1,0.9]\times[0.1,09]\times[0.1,0.9]$.

It can be noticed that the obtained in this case results are better than the results presented previously. However, the results cannot be directly compared becdause the results presented in Tables 1–4 represent the average, not the best values. The superiority of the results obtained for $\sigma^*(0)=0.4$ over the results for $\sigma^*(0)=0.1$ can be noticed once again.

We have to mention that, in general, the obtained results for the rules initiazlized using the mountain clustering technique depended heavily on the values of the

Table 5

The best results (with respect to the MSE values for the checking set) for the three-variable function modeling;
mountain clustering-based initialization, $\sigma^*(0) = 0.4$

Number of rules	(β,γ)	MSE for the training set	MSE for the checking set
4	(2.0,6.0)	4.50e-5	5.85e-5
8	(4.0,8.0)	2.49e-5	1.81e-5
16	(8.0,6.0)	6.056e-6	9.731e-6

Table 6

The best results (with respect to the MSE values for the checking set) for the three-variable function modeling;
mountain clustering-based initialization, $\sigma^*(0) = 0.1$

Number of rules	(β,γ)	MSE for the training set	MSE for the checking set
4	(2.0,4.0)	1.303e-4	1.177e-4
8	(4.0,6.0)	4.244e-5	3.146e-5
16	(2.0,2.0)	2.937e-5	6.653e-5

parameters β and γ. Moreover, some of these results were significantly worse than the average results obtained for the random and quasi-random initialization.

Within the following two examples we make use of the presented initialization techniques for electronic circuit performance modeling. The circuit performance modeling relies on a modeling of an unknown circuit performance function which depends on *a priori* chosen circuit variables. In practice the circuit performance modeling aims to speed up the optimization process of the circuit [5], [12]. To the generation of data sets a general purpose circuit simulator PSpice was used.

Example 2. Modeling of a transconductance amplifier

In this example we consider a transconductance amplifier (TCA) whose schematic diagram is presented in Fig. 2. For the investigated circuit as a performance function of interest a small-signal transconductance, g_m, was chosen. This function is defined as follows:

$$g_m = \max_{V_{in} \in [-U_m, U_m]} \left| \frac{\partial I_{load}}{\partial V_{in}} \right|$$

and it dependes (more or less) on each circuit variable.

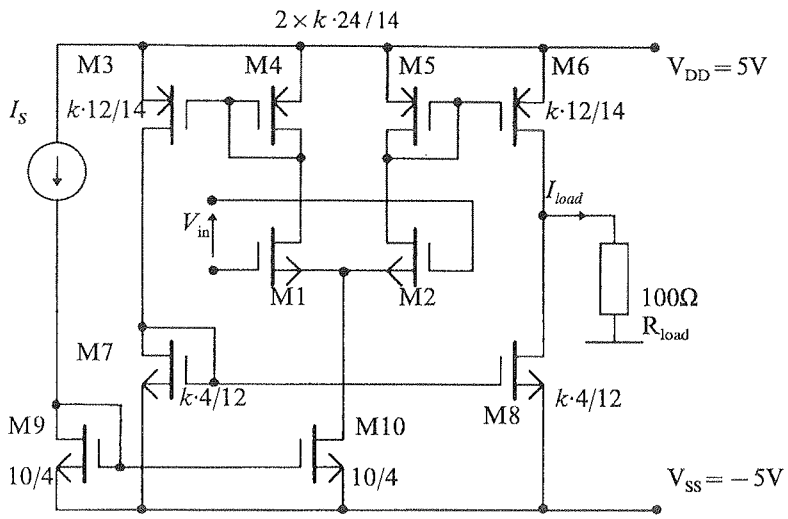


Fig. 2. Schematic diagram of the investigated TCA

For our experiments we assumed $U_m = 1V$. The chosen circuit variables (whose number has been reduced to three) and their ranges are presented in Table 7.

Table 7

Chosen circuit variables and their ranges of the investigated TCA

Circuit variable	Range
Is [μA]	10 ... 100
$(W/L)_{M1, M2}$	20/4 ... 200/4
k	0.5 ... 1.0

Based on the experimental results the parameter k has been chosen in such a way that the ciruit response $I_{load} = f(V_{in})$ remains roughly symmetrical about the point $(V_{in} I_{load}) = (0.0)$.

For the investigated circuit two data sets were generated: the training set, consisted of 100 training points, and the checking set, consisted of 100 checking points, as well. However, in this case each data set was generated using *Latin Hypercube* sampling technique [12], [13], which is a typical choice for the circuit performance modeling [5], [6], [12]. Tables 8 – 11 present, as for the previous example, the average results obtained after 10 initializations.

Based on the presented results the same conclusions as for the first example can be drown:

1. Independently of the value of $\sigma^*(0)$, the quasi-random rule initialization gives better results than the random initialization.
2. For the same initialization technique bigger value of the parameter $\sigma^*(0)$ gives better results than its smaller value.

Table 8

The average MSE values and their variances for the TCA modeling; random rule initialization, $\sigma^*(0) = 0.4$

Number of rules	MSE/variance for the training set	MSE/variance for the checking set
4	1.65e-4/2.80e-1	2.72e-4/2.70e-1
8	1.61e-5/1.63e-1	5.30e-5/2.29e-1
16	7.96e-6/7.76e-2	5.48e-5/2.04e-1

Table 9

The average MSE values and their variances for the TCA modeling; quasi-random rule initialization, $\sigma^*(0) = 0.4$

Number of rules	MSE/variance for the training set	MSE/variance for the checking set
4	7.75e-5/5.94e-1	1.13e-4/5.94e-1
8	1.69e-5/1.19e-1	4.05e-5/6.19e-2
16	1.08e-5/1.99e-1	3.74e-5/1.76e-2

Table 10

The average MSE values and their variances for the TCA modeling; random rule initialization, $\sigma^*(0) = 0.1$

Number of rules	MSE/variance for the training set	MSE/variance for the checking set
4	3.39e-4/3.60e-1	3.78e-4/2.68e-1
8	1.18e-4/3.71e-1	2.78e-4/5.11e-1
16	2.35e-5/2.04e-1	5.14e-4/3.95e-1

Table 11

The average MSE values and their variances for the TCA modeling; quasi-random rule initialization, $\sigma^*(0) = 0.1$

Number of rules	MSE/variance for the training set	MSE/variance for the checking set
4	1.82e-4/5.38e-1	2.24e-4/3.71e-1
8	4.53e-5/4.62e-1	1.677-4/1.40e-1
16	1.32e-5/2.89e-1	2.25e-4/3.06e-1

In this example a well-known phenomenon named an overfitting is observed: despite the fact that the error measure (MSE) for the training set decreases while the complexity of the model (i.e. the number of the parameters) increases, the error measure for the checking set firts decreases, but then increases. This in observed in Tables 8, 10 and 11. Note however that the use of the quasi-random initialization eliminates this effect for $\sigma^*(0) = 0.4$ (Table 9) and partly reduces it (but not eliminates) for $\sigma^*(0) = 0.1$ (Table 11).

For the sake of completeness in Tables 12 and 13 the best results obtained for the mountain clustering-based initialization are presented. Here the basis set was cnstructed as for the previous example. The presented results are generally better than for the random and quasi-random rule initialization.

Table 12

The best results (with respect to the MSE values for the checking set) for the modeling of the TCA; mountain clustering-based initialization, $\sigma^*(0) = 0.4$

Number of rules	(β,γ)	MSEfor the training set	MSE for the checking set
4	(2.0,4.0)	2.517e-05	4.661e-05
8	(6.0,6.0)	1.328e-05	3.488e-05
16	(8.0,2.0)	5.560e-06	3.100e-05

Table 13

The best results (with respect to the MSE values for the checking set) for the modeling of the TCA; mountain clustering-based initialization, $\sigma^*(0) = 0.1$

Number of rules	(β,γ)	MSEfor the training set	MSE for the checking set
4	(8.0,6.0)	7.382e-05	1.149e-04
8	(4.0,6.0)	1.686e-05	4.643e-05
16	(8.0,4.0)	1.659e-04	2.576e-04

Example 3. Modeling of an input offset voltage of an operational amplifier

As the last example an operational amplifier (OpAmp) was chosen. A schematic diagram of the circuit is presented in Fig. 3. The performance function of interest was an input offset voltage which is defined as such an input voltage, for which the output voltage equals 0V. Table 14 presents the chosen circuit variables and their ranges.

For the investigated circuit 200 training and 200 checking points were generated. As the sampling technique the *Latin Hypercube* technique was used once again.

Here the results (see Tables 15 – 18) obtained for the quasi-random initialization are only slightly better than corresponding results obtained for the random initialization.

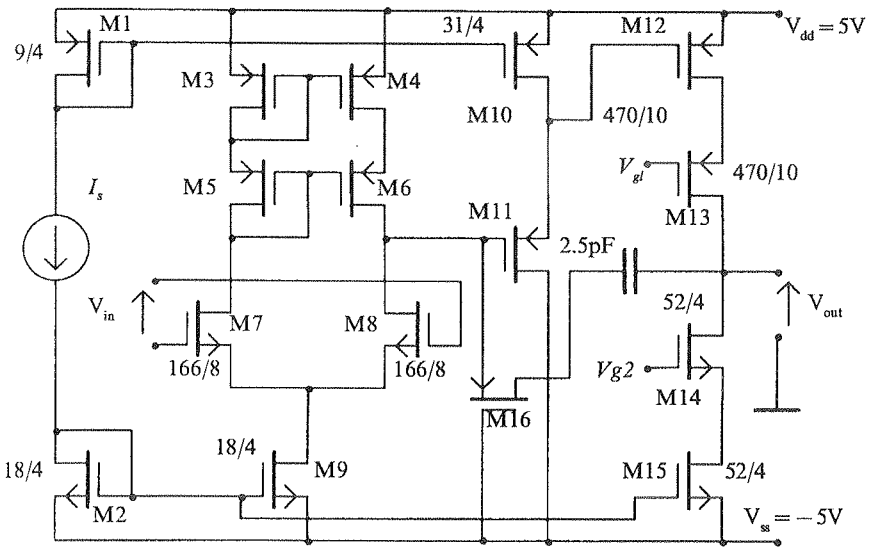


Fig. 3. Schematic diagram of the investigated operational amplifier

Table 14

Chosen circuit variable and their ranges

Circuit variable	Range
Is [μA]	10 ... 50
Vg1[V]	1.5 ... 3.0
Vg2[V]	-3.0 ... -1.5
$(W/L)_{M11}$[μm]	10/4 ... 60/4
$(W/L)_{M3,M4,M5,M6}$[μm]	10/4 ... 50/4

The overfitting appearing in this example (Table 15 and 16) is not alleviated by the quasi-random initialization. However, the superiority of the results obtained for $\sigma^*(0) = 0.4$ over the results for $\sigma^*(0) = 0.1$ can be noticed once again.

Table 15

The average MSE values and their variances for the OpAmp modeling; random rule initialization, $\sigma^*(0) = 0.4$

Number of rules	MSE/variance for the training set	MSE/variance for the checking set
4	1.19e-4/2.82e-1	1.71e-4/1.78e-1
8	3.58e-5/1.28e-1	9.18e-5/7.25e-2
16	1.86e-5/1.23e-1	1.41e-4/4.99e-2

Table 16

The average MSE values and their variances for the OpAmp modeling; quasi-random rule initialization, σ^* (0) = 0.4

Number of rules	MSE/variance for the training set	MSE/variance for the checking set
4	8.79e-5/3.56e-1	1.14e-4/3.79e-1
8	3.13e-5/7.27e-2	8.22e-5/1.45e-1
16	1.64e-5/4.53e-2	1.03e-4/1.50e-1

Table 17

The average MSE values and their variances for the TCA modeling; random rule initialization, σ^* (0) = 0.1

Number of rules	MSE/variance for the training set	MSE/variance for the checking set
4	1.01e-3/4.38e-1	9.85e-4/3.87e-1
8	1.97e-4/1.51e0	2.93e-4/1.07e0
16	9.88e-5/2.74e-1	2.33e-4/1.38e-1

Table 18

The average MSE values and their variances for the TCA modeling; quasi-random rule initialization, σ^* (0) = 0.1

Number of rules	MSE/variance for the training set	MSE/variance for the checking set
4	3.15e-4/1.77e0	3.44e-4/1.38e0
8	1.22e-4/1.75e-1	1.64e-4/1.48e-1
16	4.68e-5/2.35e-1	1.64e-4/4.22e-2

Table 19

The best results (with respect to the MSE values for the checking set) for the modeling of the OpAmp; mountain clustering-based initialization, σ^* (0) = 0.4

Number of rules	(β,γ)	MSE for the training set	MSE for the checking set
4	(8.2),(8.4) (8.6),(6.2),(6.4)	4.98e-5	6.39e-5
8	(6.0,6.0)	2.22e-5	6.06e-5
16	(2.0,2.0)	1.46e-5	5.74e-5

Table 20

The best results (with respect to the MSE values for the checking set) for the modeling of the OpAmp; mountain clustering-based initialization, $\sigma^*(0) = 0.1$

Number of rules	(β,γ)	MSE for the training set	MSE for the checking set
4	(4.0,4.0)	6.19e-5	7.94e-5
8	(4.0,2.0)	5.17e-5	7.15e-5
16	(4.0,2.0)	4.93e-5	7.48e-5

The presented in Tables 19 and 20 results show that the mountain clustering initialization can give better results than the remainder two methods. Note however that if we wish to construct the basis set in a manner described previously, we would have to generate $5^6 = 15626$ basis points, and the construction of the mountain function could prove computationally expensive in this case. Therefore, for high dimensional cases, we propose to use the same sampling technique as for the training (and checking) set genration. In this example as a basis set we used 200 basis points generated in the input-output spece using *Latin Hypercube* sampling technique.

5. CONCLUSIONS AND FINAL REMARKS

In the paper three methods of the neurofuzzy model parameter initialization were experimentally compared. Our experiments shown that, in general, the simple quasi-random initialization technique overperforms the random initialization. Moreover, this first technique can reduce or even eliminate an appearance of the overfitting to the training data. We has shown, as well, that it is better to pick up the larger initial value of the parameter σ^*. We believe that the larger value of this parameter improves generalization abilities of the model: for larger value of this parameter the values of the membership functions change more smoothly along each interval X_i, which ensures that the output of the model changes more smoothly, as well. It helps to model performance functions which, as it was shown in [14], are rather monotonous and do not chnge rapidly. It can be believed that if we deal with strongly nonlinear functions, smaller initial values of σ^* will give better results than its bigger initial values.

The proposed in this paper quasi-random initialization is probably the simplest way of using the training points. In order to improve this method, one could check a distance between randomly chosen training points: if the distance between any two chosen points is small it means that these two points will initialize similar rules, and that one of this rules will be superfluous. One can invent many heuristic versatile methods which would be able to eliminate superfluous rules.

Finally, the mountain clustering-based initialization can give very good results in some cases. Unfortunately, the obtained results strongly depend on the values of two parameters which control the mountain function construction and degradation. We have not managed to find the best values of this parameters. Probably more experimental or theoretical works have to be performed in order to find the optimal values of these parameters.

REFERENCES

1. D. Rutkowska: *Inteligentne systemy obliczeniowe.* Warszawa, Akademicka Oficyna Wydawnicza PLJ, 1997.
2. J-S. R. Jang: *ANFIS Adaptive-Netwoerk Based Fuzzy Inference System.* IEEE Trans. Syst., Man and Cybernetics, Vol. 23, No. 3, May/June 1993.
3. R. Horbowski, M. Białko: *Genetic Algorithm-Based Structure Identufucation of an Associative Memory Network.* Proc. of the XXIInd Conf. on Circ. Theory and Electr. Networks, Warszawa, Stare Jabłonki, October 1999.
4. K. M. Bossley: *Meurofuzzy Modelling Approaches in System Identification.* PhD Thesis, University of Southampton, UK, 1997.
5. A. Torralba, J. Chavez, L. Franquelo: *Circuit Performance Modelling by Means of Fuzzy Logic.* IEEE Trans on CAD, Vol. 15, No 11, pp. 1391–1398, Nov. 1996.
6. R. Horbowski, M. Białko: *Neurofuzzy B-spline-Based Circuit Performance Modeling Aided by Means of Genetic Algorithm.* Proc. of the XXIInd Conf. on Circ. Theory and Electr. Networks. Warszawa Stare Jabłonki, October 1999.
7. R. R. Yager, D. P. Filev: *Podstawy modelowania i sterowania rozmytego*, Warszawa, WNT, 1995.
8. M. Brown, C. Harris: *Neurofuzzy Adaptative Modeling and Control.* Prentice Hall, Hemel Hempstead, UK, 1994.
9. D. Rutkowska, M. Piliński, L. Rutkowski: *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte.* Warszawa-Łódź, PWN, 1997.
10. R. Jager: *Fuzzy Logic in Control.* PhD Thesis, Technische Universiteit Delft, Holland, 1995.
11. D. Drainkov, H. Hellendoorn, M. Reinfrank: *Wprowadzenie do sterowania rozmytego*, Warszawa, WNT, 1996.
12. M. A. Styblinski, S. Aftab: *Combination of Interpolation and Self-Organizing Approximation Techniques — A New Approach to Circuit Performance Modelling.* IEEE Trans. on CAD, Vol. 12, No 11, pp. 1775–1785, Nov. 1993.
13. M. D. McKay, R. J. Beckman, W. J. Conover: *A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Outpu from a Computer Code.* Technometrics, Vol. 21, No. 2, pp. 239–245, 1979.
14. R. Horbowski, M. Biłko: *The Generation of Training Data Points for Ciruit Performance Modelling by Means of Fuzzy System.* Kwartalnik Elektroniki i Telekomunikacji PAN, Vol. 44, Nr. 4, ss.471–486, 1998.

R. HORBOWSKI

PORÓWNANIE TRZECH METOD DLA MODELI ROZMYTYCH NEURONÓW I ICH APLIKACJI W OBWODACH JAKOŚCIOWEGO MODELOWANIA

Strzeszczenie

W pracy zaprezentowano i eksperymentalnie porównano trzy metody inicjacji parametrów modeli neurorozmytych. Pierwszą z rozważanych metod jest charakterystyczna dla sieci neuronowych inicjacja losowa. Wyniki uzyskane na bazie tej metody są porównywane z wynikami uzyskanymi z wykorzystaniem tzw. metody inicjacji pseudo-losowej. Ta druga, alternatywna metoda wykorzystuje losowo wybrane ze zbioru uszącego punkty uczące, które inicjują parametry odpowiedzialne za rozmieszczenie funkcji przynależności definiujących poszczególne etykiety lingwistyczne występujące w regułach rozmytych. Podejście takie jest możliwe, ponieważ wspomiane parametry posiadają swoją interpretację fizyczną.

Prezentowane wyniki pokazują, iż średnie wyniki uzyskane metodą inicjaji pseudo-losowej przewyższają swoje odpowiedniki uzyskane metodą inicjacji losowej. Co więcej, występujące dla modeli posiadających dużą liczbę parametrów ponaddopasowanie do danych uczących, może zostać częściowo zredukowane lub nawet wyeliminowane przy użyciu metody pseudo-losowej.

Trzecią metodą rozważaną w pracy jest metoda oparta na górskim grupowaniu danych. Eksperymenty pokazały, iż metoda ta może dać lepsze wyniki niż metody wspomiane powyżej. Należy jednak podkreślić, iż jakość wyników uzyskiwanych tą metodą silnie zależy od wartości dwóch parametrów, które nią sterują. Niestety, mimo przeprowadzenia licznych prób nie udało się ustalić optymalnych wartości tych parametrów.

Prezentowana praca porusza także niektóre aspekty i problemy związane z prezentowaną techniką modelowania. Sugerowane jest również udoskonalenie proponowanej metody pseudolosowej.

Słowa kluczowe: modelowanie rozmyte i neurorozmyte, estymacja i inicjacja parametrów, modelowanie adaptacyje, modelowanie funkcji układowych układów elektronicznych.

Heu
Boole
prze

W
dekom
Curtis'
Algory
ekspery
naniu z

Słowa

Jednym
struktur pr
które realiz
logicznych,
Proste
PLD. Zawi
żadnych o
Wynika to
dołączenia
dowolnej b
struktur, w
prostych uk
odpowiedni
odpowiada

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2000, 46, z. 2, ss. 195–210

Heurystyczna metoda dekompozycji zespołu funkcji Boole'owskich wykorzystująca dekompozycje złożone, przeznaczona dla układów FPGA typu tablicowego

DARIUSZ KANIA

Instytut Elektroniki, Politechnika Śląska

Otrzymano 1999.11.10
Autoryzowano 2000.04.25

W artykule przedstawiono metodę dekompozycji opartą na odpowiednim doborze dekompozycji złożonych zespołu funkcji Boole'owskich. Podstawą metody jest teoria Curtis'a. Głównym elementem algorytmu jest analiza struktury wierszy tablicy podziałów. Algorytm został opracowany w postaci eksperymentalnego programu Decomp. Wyniki eksperymentów potwierdziły efektywność przedstawionej metody dekompozycji w porównaniu z innymi poprzednio opublikowanymi.

Słowa kluczowe: synteza logiczna, dekompozycja, podział, FPGA

1. WPROWADZENIE

Jednym z podstawowych problemów syntezy układów cyfrowych na bazie struktur programowalnych stał się sposób podziału projektu na odpowiednie części, które realizowane są w odrębnych układach PLD lub konfigurowalnych blokach logicznych, występujących w strukturach CPLD lub FPGA [2,32].

Proste programowalne układy logiczne należą do najpopularniejszych układów PLD. Zawierają one niewielką liczbę elementów logicznych, lecz nie wprowadzają żadnych ograniczeń związanych z możliwościami połączeń wewnątrz struktury. Wynika to z dużej elastyczności struktury macierzowej, zapewniającej możliwość dołączenia wszystkich sygnałów wejściowych i wyjściowych (sprzężenie zwrotne) do dowolnej bramki OR zawartej w układzie. Ze względu na niewielką złożoność tych struktur, w przypadku większych systemów zachodzi konieczność tworzenia sieci prostych układów PLD. Głównym problemem syntezy w tym przypadku jest dobór odpowiednich struktur PAL, PLA oraz problem podziału projektu na bloki, odpowiadające poszczególnym układom [10, 14].

W przypadku wykorzystania rozbudowanych struktur macierzowych (CPLD) możliwa jest realizacja całego projektu w jednym układzie. Podstawowym problemem syntezy w tym przypadku jest podział projektu na poszczególne bloki logiczne zawarte wewnątrz struktury CPLD. W większości przypadków bloki logiczne zawarte w strukturach CPLD wykorzystują bazową strukturę charakteryctyczną dla układów PAL (programowalna matryca AND, stałe połączenia w matrycy OR), stąd równie ważnym problemem sysntezy jest problem efektywnego wykorzystania termów [5, 15] i uwzględnienia zasobów wewnętrznych bloku, scharakteryzowanych na przykład poprzez jego pojemność kodową [16].

Odmienną architekturę wewnętrzną mają układy FPGA typu tablicowego (ang. Look-up Table). Składają się one z matrycy konfigurowalnych komórek otoczonych na obrzeżach blokami wejścia/wyjścia. Pomiędzy rzędami i kolumnami komórek znajdują się kanały z liniami połączeń. Możliwość prowadzenia połączeń jedynie w wydzielonych do tego celu kanałach sprawia, że niezwykle ważnym etapem syntezy stał się sposób rozmieszczenia bloków wewnątrz struktury (ang. placement) oraz sposób prowadzenia połączeń (ang. routing). Ponieważ zasoby logiczne konfigurowalnych bloków logicznych umożliwiają realizację dowolnej funkcji o określonej liczbie argumentów stąd głównym problemem syntezy jest dekompozycja prowadząca do rozbicia całego projektu na poszczególne części o zadanej liczbie wejść.

Jak widać z przedstawionych rozważań proces projektowania układów cyfrowych w oparciu o struktury programowalne wiąże się bezpośrednio z dekompozycją, stanowiącą strategiczny etap całej syntezy. Sercem wielu systemów (przed wszystkim uniwersyteckich) wspomagających projektowanie układów cyfrowych na bazie struktur programowalnych jest dekompozycja najczęściej ukierunkowana na wielopoziomową realizację układów w strukturach bramkowych lub strukturach typu tablicowego. Pierwsze algorytmy syntezy (MIS-PGA [24, 26], ASYL [1, 4], Chortle [7]) przeznaczone dla struktur bramkowych (ang. gate array) zostały również wykorzystane w procesie syntezy układów typu tablicowego. Algorytmy te oparte są na faktoryzacji wyrażeń boolowskich wspomaganej procedurami grupowania, łączenia funkcji w grupy zależne od liczby wyjść bloków logicznych, leksykograficznego porządkowania zmiennych, programowaniem dynamicznym itd. Bardzo często poszczególne etapy dekompozycji polegają na iteracyjnym podziale sieci logicznej, odpowiadającej w początkowym etapie wyrażeniu Boole'owskiemu po dwupoziomowej minimalizacji. Podział sieci logicznej polega na odpowiednim wyborze węzłów, gałęzi itp. [6, 7, 23].

Klasyczny model dekompozycji funkcjonalnej zaproponowany przez Ashenhurst'a [3] i rozszerzony przez Curtis'a [12, 13] stanowi podstawę drugiej grupy algorytmów dekompozycji [8, 17, 21, 27, 33]. Algorytmy te wykorzystują często różnorodne strategie kolorowania grafów [28, 31, 34], transformacji funkcji [20, 34], łączenia klasycznej dekompozycji z procedurami podziału wyjść [17, 25] itd. Wydaje się, że jedną z zalet tych metod syntezy jest uwzględnianie stanów nieokreśloności w początkowym etapie syntezy najczęściej związanym z dekompozycją. W ostatnich latach dużym zainteresowaniem cieszą się metody wykorzystujące binarne diagramy decyzyjne [9, 19, 20, 30].

Celem niniejszego artykułu jest przedstawienie wyników dekompozycji prowadzącej do realizacji układów cyforwych w strukturach FPGA typu tablicowego, uzyskanych za pomocą systemu Decomp. System ten bazuje na klasycznym modelu dekompozycji funkcjonalnej i poszukując odpowiednich dekompozycji złożonych (iteracyjnych, wielokrotnych) umożliwia realizację funkcji logicznych na konfigurowalnych blokach logicznych typu tablicowego o zadanej liczbie wejść i wyjść.

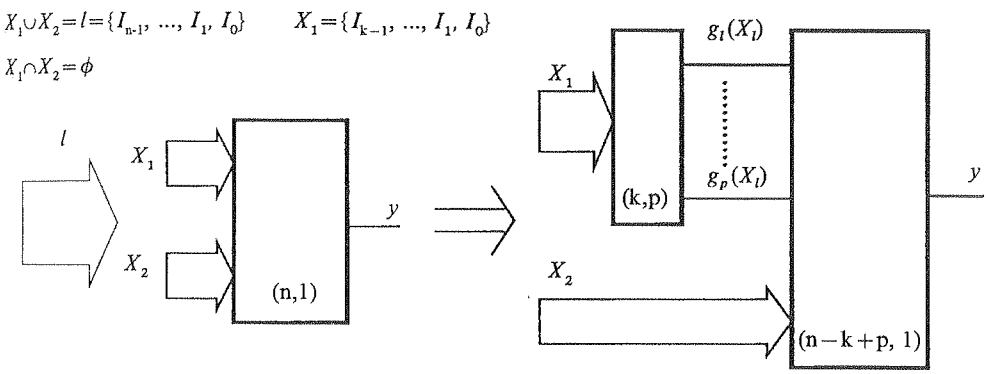
2. PODSTAWY TEORETYCZNE

Funkcja $y=f(I_{n-1},\ldots,I_1, I_0)=f(X_2, X_1)$ podlega dekompozycji prostej tzn. $f(X_2, X_1)=F[g(X_1)X_2]$ wtedy i tyklo wtedy, gdy złożoność kolumnowa matrycy podziałów $\nu(X_2 \mid X_1)\leqslant 2$ [3]. Zbiory X_1 i X_2 nazywane są odpowiednio zbiorem związanym i wolnym, przy czym $X_1\cup X_2=I$ oraz $X_1\cap X_2=\phi$, gdzie $I=\{I_{n-1},\ldots,I_1, I_0\}$.

Twierdzenie o dekompozycji prostej zostało rozszerzone [12] ponieważ

$$\nu(X_2 \mid X_1)\leqslant 2^p \Leftrightarrow f(X_2, X_1)=F[g_1(X_1),g_1(X_1),\ldots,g_p(X_1),X_2] \quad (1)$$

Wykorzystanie takiego modelu dekompozycji odpowiada rozbiciu bloku scharakteryzowanego uporządkowaną parą liczb (liczba wejść, liczba wyjść) wynoszącą $(n, 1)$ na dwa inne o parametrach (k, p) i (n−k+p,1), przedstawione na rys. 1.



Rys. 1. Zasada rozłącznej dekompozycji Curtis'a i odpowiadające jej rozbicie układu

Celem dekompozycji funkcji jest odpowiadające jej rozbicie projektowanego układu logicznego na podukłady o zadanej liczbie wejść i wyjść (realizacja układu cyfrowego w strukturach FPGA typy tablicowego zawierającego konfigurowalne bloki logiczne). Rozbicie układu pociąga za sobą jednak ekspansję całkowitej liczby wyjść. Z punktu widzenia projektanta najważniejsze są dwa parametry charakteryzujące układ po dekompozycji:

- przyrost sumarycznej liczby wyjść,
- maksymalna liczba wejść bloków logicznych, które powinny być wykorzystane do realizacji podukładów powstających po dekompozycji.

Niech α będzie liczbą równą przyrostowi liczby wyjść układu po dekompozycji, natomiast ß-parametrem określającym maksymalną liczbę wejść podukładów po dekompozycji.

Dla przedstawionego modelu dekompozycji Curtis'a (1) parametr $\alpha=p$, natomiast $\beta=\max(\overline{X1}, \overline{X2}+p)$. Efektem dekompozycji jest podział zbioru argumentów funkcji $I=\{I_{n-1},\ldots,I_1 I_0\}$ na dwa podzbiory $X_1=\{I_{k-1},\ldots,I_1, I_0\}$ i $X_2=\{I_{n-1},\ldots,I_{k+1},I_k\}$, takie, że $X_1 \cup X_2 = I$ oraz $X_1 \cap X_2 = \phi$.

Definicja 1:

Niech P_i będzie podziałem zbioru argumentów $I=\{I_{n-1},\ldots,I_1,I_0\}$ na dwa podzbiory $X_{1i}=\{I_{k-1},\ldots,I_1, I_0\}$ i $X_{2i}=\{I_{n-1},\ldots,I_{k+1}, I_k\}$.

Podział P_i implikowany przez dekompozycję funkcji $y=f(I_{n-1},\ldots,I_1,I_0)$, taką, że $f(I_{n-1},\ldots,I_1 I_0)=f(X_{2i}, X_{1i})=F[g_1(X_{1i}),\ldots,g_p(X_{1i}), X_{2i}]$ prowadzi do ograniczenia liczby wejść podukładów po podziale (w stosunku do układu przez podziałem) wtedy, gdy $I = n > \beta = \max(\overline{X1}, \overline{X2}+p)$.

Jeżeli istnieje kilka różnych (co najmniej dwa) podziałów argumentów P_i, dla których zachodzi dekompozycja zgodnie z zależnością (1) to istnieje możliwość poszukiwania dekompozycji wielokrotnych (DW) i/lub iteracyjnych (DI).

W pracy [13] można znaleźć różne modele dekompozycji wielokrotnej i iteracyjnej. Przedstawione tam twierdzenia można uogólnić i podać w postaci uogólnionego twierdzenia o dekompozycji wielokrotnej i dekompozycji iteracyjnej.

Uogólnione twierdzenie o dekompozycji wielokrotnej:

Jeżeli

$$y=F_1[G_1(X_{11}), X_{12}, X_{13},\ldots X_{1l}, X_2]$$

$$y=F_2[X_{11}, G_2(X_{12}), X_{13},\ldots X_{1l}, X_2]$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$y=F_l[X_{11}, X_{12}, X_{13},\ldots G_l(X_{1l}), X_2]$$

gdzie

$$G_1(X_{11})=[g_{11}(X_{11}), g_{12}(X_{11}), \ldots g_{1p1}(X_{11})]$$

$$G_2(X_{12})=[g_{21}(X_{12}), g_{22}(X_{12}), \ldots g_{2p2}(X_{12})]$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

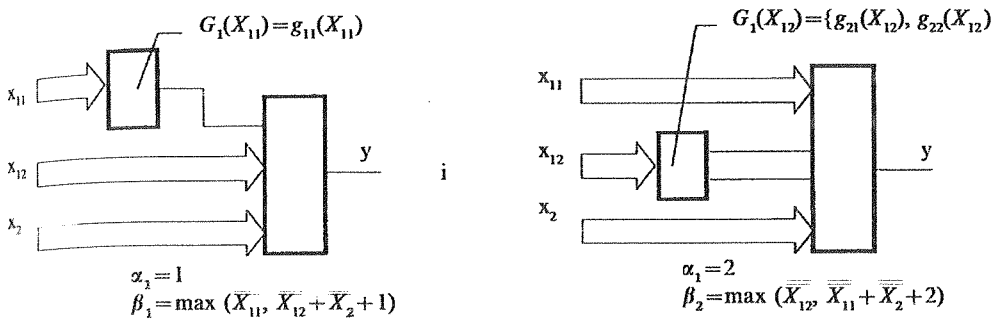
$$G_l(X_{1l})=[g_{l1}(X_{1l}), g_{l2}(X_{1l}), \ldots g_{lpl}(X_{1l})]$$

to

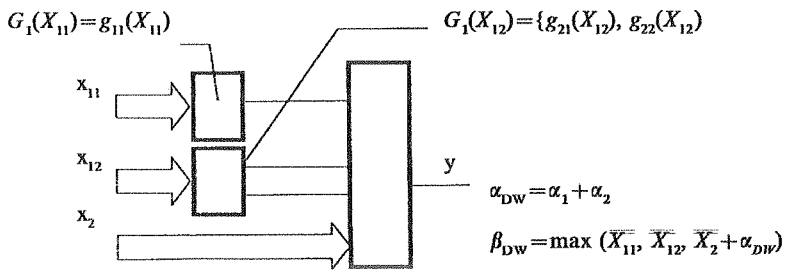
$$y=H[G_1(X_{11}), G_2(X_{12}), \ldots G_l(X_{1l}), X_2]$$

Istota twierdzenia oraz wartości współczynników charakteryzujące układy po dekompozycji przedstawione są na rys. 2.

Jeżeli



to



Rys. 2. Podział układu — dekompozycja wielokrotna funkcji

W podobny sposób można przedstawić uogólnione twierdzenie o dekompozycji iteracyjnej.

Jeżeli

$$y=F_1[G_1(X_{11}), X_{12}, X_{13},\ldots X_{1l}, X_2]$$

$$y=F_2[G_2(X_{11}, X_{12}), X_{13},\ldots X_{1l}, X_2]$$

$$y=F_3[G_3(X_{11}, X_{12}, X_{13}),\ldots X_{1l}, X_2]$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$y=F_l[G_l(X_{11}, X_{12}, X_{13},\ldots X_{1l}), X_2]$$

gdzie

$$G_1(X_{11})=[g_{11}(X_{11}), g_{12}(X_{11}), \ldots g_{1p_1}(X_{11})]$$

$$G_2(X_{11}, X_{12})=[g_{21}(X_{11}, X_{12}), g_{22}(X_{11}, X_{12}), \ldots g_{2p_2}(X_{11}, X_{12})]$$

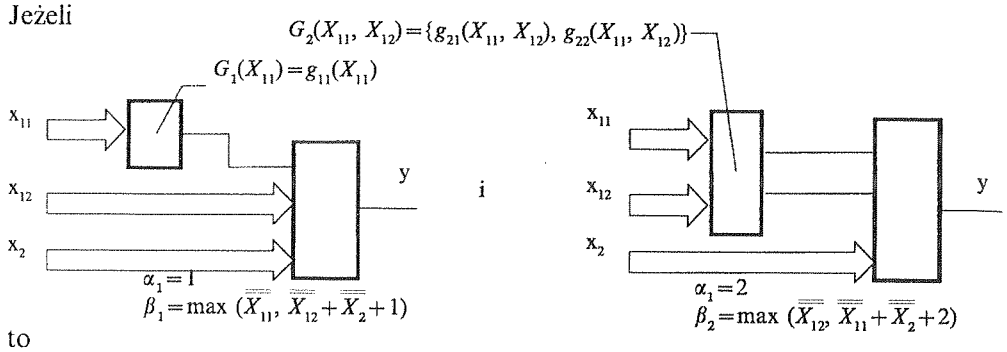
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$G_l(X_{11},\ldots,X_{1l})=[g_{l1}(X_{11},\ldots X_{1l}), g_{l2}(X_{11},\ldots X_{1l}), \ldots g_{lp_l}(X_{11},\ldots X_{1l})]$$

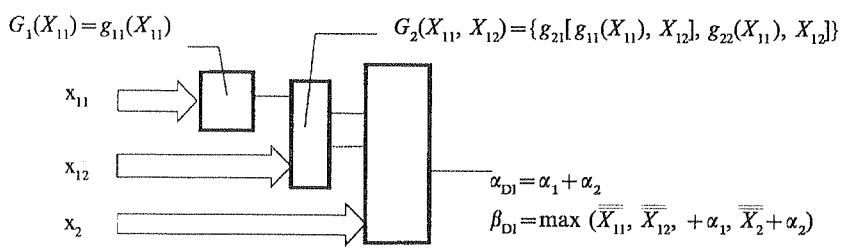
to

$$y=F_l[G_l[\ldots G_2[G_1(X_{11}), X_{12}], \ldots], X_{1l}], X_2]$$

Jeżeli



to



Rys. 3. Podział układu — dekompozycja inetracyjna funkcji

Istota twierdzenia wraz z wartościami współczynników charakteryzujących układy po dekompozycji przedstawiona jest na rys. 3.

Z przedstawionych uogólnionych twierdzeń o dekompozycji wielokrotnej (DW) i dekompozycji iteracyjnej (DI) wynika, że funkcja podlega DW lub DI wtedy, gdy istnieją co najmniej dwie odpowiednie dekompozycje funkcji. W celu przedstawienia algorytmu poszukiwania DW i DI konieczne staje się wprowadzenie dodatkowych określeń.

Definicja 2:

Tablicą podziałów funkcji $y=f(I_{n-1},...,I_1,I_0)$ nazywamy tablicę o wymiarach $k \times n$, której wierszom przyporządkowane są podziały P_1, P_2,...,P_k, dla których $\max(X1, X2+p)<n$, kolumnom — argumenty funkcji, natomiast elementami tablicy są „*", umieszczone w kolumnach odpowiadających zmiennym, należącym do zbiorów związanych poszczególnych podziałów.

Definicja 3:

Wiersze i, j tablicy podziałów nazywamy rozłącznymi wtedy i tylko wtedy, gdy odpowiadające im zbiory związane spełniają zależności $X_{1i} \cap X_{1j}=\phi$. (tzn.: „*" występują w różnych kolumnach).

Definicja 4:

Wiersz i-ty podziałów dominuje nad wierszem j-tym wtedy i tylko wtedy, gdy $X_{1j} \supset X_{1i}$. (tzn.: „*" w wierszu j są podzbiorem „*" w wierszu i)

Przykład 1:

	I6	I5	I4	I3	I2	I1	I0
P_1	*	*	*	*		*	
P_2		*		*		*	
P_3			*		*		*
P_4	*			*		*	

wiersz 1 dominuje nad wierszem 2 i 4

wiersz 2 jest rozłącznym z wierszem 3

wiersz 3 jest rozłączny z wierszem 4

Bazując na określeniach zawartych w definicjach 2, 3 można twierdzenie o dekompozycji wielokrotnej przedstawić następująco:

Twierdzenie 1:

Funkcja y opisana na zbiorze $I=\{I_{n-1},\ldots,I_1,I_0\}$ podlega dekompozycji wielokrotnej wtedy, gdy spośród k wierszy tablicy podziałów można wybrać s wierszy wzajemnie rozłącznych.

Dowód twierdzenia wynika bezpośrednio z uogólnionego twierdzenia o dekompozycji wielokrotnej oraz definicji 2 i 3.

Oznaczmy wzajemnie rozłączne wiersze tablicy podziałów jako P_1, P_2,..., P_s, odpowiadające im zbiory związane odpowiednio X_{11}, X_{12}, X_{1s} i wyznaczmy na ich podstawie współczynniki α_1, α_2,..., α_s oraz β_1, β_2, ..., β_s. Znając współczynniki α_1, α_2,..., α_s oraz β_1, β_2, ..., β_s można wyznaczyć współczynniki α i β dla dekompozycji wielokrotnej, oznaczane jako α_{DW} oraz β_{DW}.

Współczynniki α_{DW}, β_{DW} wynoszą odpowiednio:

$$\alpha_{DW}=\sum_{i=1}^{s}\alpha_i$$

$$\beta_{DW}=\max(\overline{\overline{X}}_{11},\ \overline{\overline{X}}_{12},\ldots,\overline{\overline{X}}_{1s},\ \overline{\overline{X}}_2+\alpha_{DW})$$

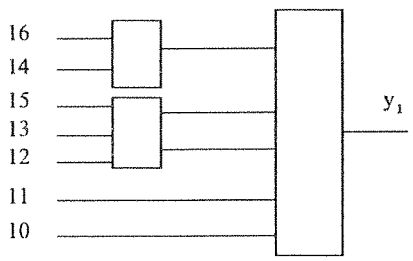
Przykład 2:

Niech przykładowa funkcja y_1 podlega dekompozycjom, które prowadzą do podziałów argumentów przedstawionych w wierszach P_1 i P_2 tablicy podziałów.

	I6	I5	I4	I3	I2	I1	I0	α_i	β_i	
P_1	*		*					1	6	$X_{11}=\{I6, I4\}$
P_2		*		*	*			2	6	$X_{12}=\{I5, I3, I2\}$

Ponieważ wiersze P_1 i P_2 tablicy podziałów są rozłączne stąd funkcję y_1 można zrealizować w sposób przedstawiony na rys. 4.

Bazując na określeniach zawartych w definicjach 2, 4 twierdzenie o dekompozycji iteracyjnej można przedstawić w następującej formie:



$$X_{11} = \{I6, I4\}$$
$$X_{12} = \{I5, I3, I2\}$$
$$X_2 = \{I1, I0\}$$

$$\alpha_{DW} = 3$$
$$\beta_{DW} = \max(\overline{\overline{X_{11}}}, \overline{\overline{X_{12}}}, \overline{\overline{X_2}} + \alpha_{DW})$$
$$\beta_{DW} = max\ (2, 3, 5) = 5$$

Rys. 4. Dekompozycja wielokrotna funkcji y_1

Twierdzenie 2:

Funkcja y opisana na zbiorze $I=\{I_{n-1},\ldots,I_1,I_0\}$ podlega dekompozycji iteracyjnej wtedy, gdy spośród k wierszy tablicy podziałów można wybrać s wierszy takich, że wiersz s dominuje nad wierszem $(s-1)$, $(s-2)$ nad $(s-2)$,..., wiersz drugi nad wierszem pierwszym.

Dowód twierdzenia wynika bezpośrednio z uogólnionego twierdzenia o dekompozycji iteracyjnej oraz *definicji* 2 i 4.

Oznaczamy dominujące wiersze tablicy podziałów jako $P_1, P_2,\ldots, P_s$ odpowiadające im zbiory związane odpowiednio $X_{11}, X_{12},\ldots, X_{1s}$ i wyznaczamy na ich podstawie współczynniki $\alpha_1, \alpha_2,\ldots, \alpha_s$ oraz $\beta_1, \beta_2,\ldots, \beta_s$. Znając współczynniki $\alpha_1, \alpha_2,\ldots, \alpha_s$ oraz $\beta_1, \beta_2,\ldots, \beta_s$ można wyznaczyć współczynniki α i β dla dekompozycji iteracyjnej, oznaczane jako α_{DI} oraz β_{DI}.

Współczynniki α_{DI}, β_{DI} wynoszą odpowiednio:

$$\alpha_{DI} = \sum_{i=1}^{s} \alpha_i$$

$$\beta_{DI} = \max(\overline{\overline{X_{11}}}, \overline{\overline{X_{12}}} - \overline{\overline{X_{11}}} + \alpha_1, \ldots, \overline{\overline{X_{1s}}} - \overline{\overline{X_{1(s-1)}}} + \alpha_{s-1}, \overline{\overline{X_2}} + \alpha_{DI})$$

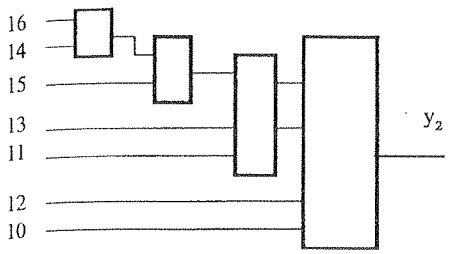
Przykład 3:

Niech przykładowa funkcja y_2 podlega dekompozycjom, które prowadzą do podziałów argumentów przedstawionych w wierszach P_1, P_2 i P_3 tablicy podziałów.

	I6	I5	I4	I3	I2	I1	I0	α_i	β_i	
P_1	*	*	*	*		*		1	6	$X_{11}=\{I6, I5, I4, I3, I1\}$
P_2		*		*		*		1	5	$X_{12}=\{I5, I3, I1\}$
P_3				*		*		2	5	$X_{13}=\{I3, I1\}$

Ponieważ wiersz P_1 tablicy podziałów dominuje nad wierszem P_2 oraz wiersz P_2 dominuje nad wierszem P_3 stąd możliwa jest dekompozycja i realizacja funkcji y_2 przedstawiona na rys. 5.

Jak widać z przykładu 1 i 2 istota poszukiwania dekompozycji złożonych funkcji polega na analizie wierszy tablicy podziałów.



$X_{11} = \{I6, I5, I4, I3, I1\}$
$X_{12} = \{I5, I3, I1\}$
$X_{13} = \{I3, I, 1\}$
$X_2 = \{I2, I0\}$

$\alpha_{DI} = 4$
$\beta_{DI} = \max(2, 2, 3, 4) = 4$

Rys. 5. Dekompozycja interacyjna funkcji y_2

Przykład 4:

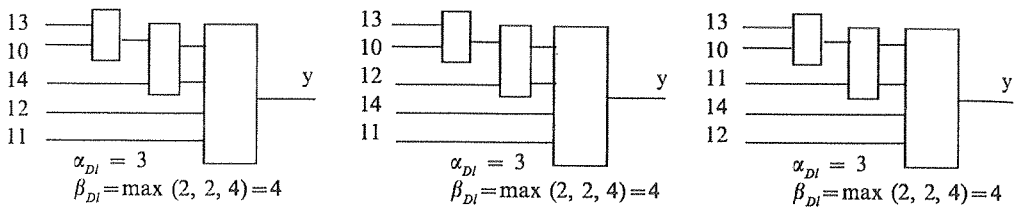
Znajdźmy wszystkie dekompozycje złożone przykładowej funkcji opisanej w formacie Berkeley'owskim *y.pla*, zakładając, że tablicę podziałów tworzymy poszukując dekompozycje, dla których $X_1 \leqslant 3$.

```
.i5
.o1
.ilb I4,I3,I2,I1,I0
.ob y
.p 12
00000 1
00011 1
00110 1
00111 1
01010 1
01011 1
01110 1
01111 1
10000 1
10101 1
11100 1
11101 1
.e
```

W wyniku poszukiwania podziałów uzyskujemy tablicę podziałów wraz z współczynnikami α i β.

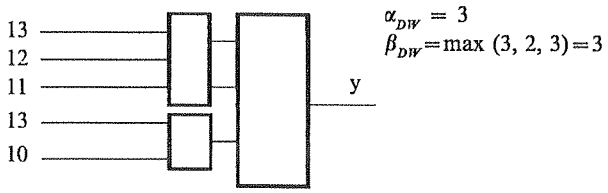
	I4	I3	I2	I1	I0	α_i	β_i
P_1		*			*	1	4
P_2	*	*			*	2	4
P_3	*		*	*		2	4
P_4		*	*		*	2	4
P_5		*		*	*	2	4

1. Szukamy wierszy dominujących tablicy podziałów i na tej podstawie znajdujemy trzy dekompozycje iteracyjne (rys. 6).



Rys. 6. Realizacje funkcji y po dekompozycjach interacyjnych

2. Szukamy wierszy wzajemnie rozłącznych i na tej podstawie znajdujemy dekompozycję wielokrotną (rys. 7).



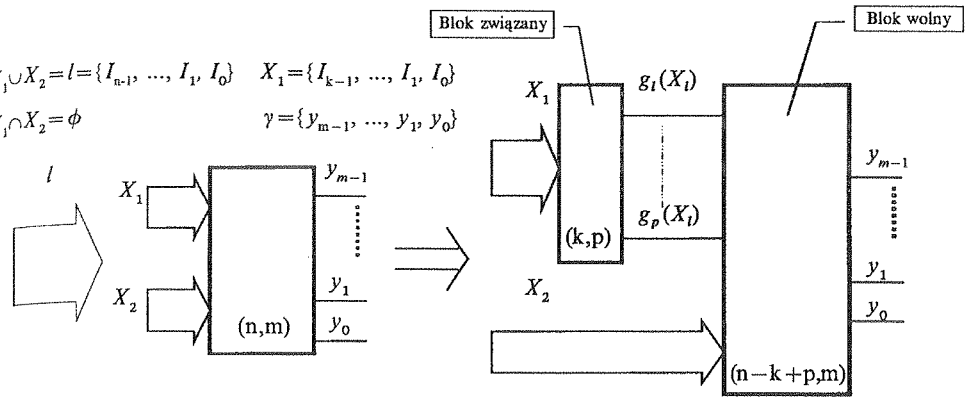
Rys. 7. Realizacje funkcji y po dekompozycji wielokrotnej

Dekompozycja wielokrotna umożliwia bezpośrednią realizację funkcji w oparciu o 3-wejściowe konfigurowalne bloki logiczne typu tablicowego przedstawioną na rys. 7.

3. DEKOMPOZYCJE ZESPOŁU FUNKCJI

W przypadku realizacji zespołu funkcji istnieje możliwość równoczesnego rozpatrywania matrycy podziałów całego zespołu lub wybranego podzbioru funkcji. Wykorzystanie takiego modelu dekompozycji [34] odpowiada rozbiciu wielowyjściowego bloku scharakteryzowanego uporządkowaną parą liczb (liczba wejść, liczba wyjść) wynoszącą $(n, \overline{Y})$ na dwa inne o parametrach (k, p) i $(n-k+p, \overline{Y})$, gdzie $Y=\{y_{m-1}\ldots y_1 y_0\}$. Rozbicie układu odpowiadające dekompozycji zespołu funkcji przedstawiono na rys. 8.

Twierdzenie o dekompozycji wielokrotnej i iteracyjnej, przedstawione dla funkcji jednowejściowej można bezpośrednio uogólnić na zespół funkcji. Poszukiwanie odpowiedniej dekompozycji złożonej zespołu funkcji zaimplementowane w systemie Decomp opiera się na analizie wierszy tablicy podziałów, która opisuje podziały prowadzące do ograniczenia liczby wejść podukładów tzn. $\beta\text{-max}(\overline{X1},\overline{X2}+p)>n=\overline{I}$, gdzie I jest zbiorem argumentów zespołu funkcji X1 (X2) — zbiorem związanym (wolnym) analizowanego zespołu funkcji.



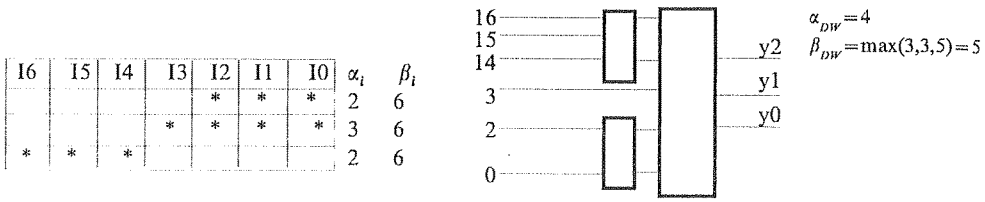
Rys. 8. Zasada rozłącznej dekompozycji zespołu funkcji i odpowiadające jej rozbicie układu

4. STRATEGIA POSZUKIWANIA DEKOMPOZYCJI UMOŻLIWIAJĄCĄ REALIZACJĘ ZESPOŁU FUNKCJI NA BLOKACH ZAWIERAJĄCYCH n_b-WEJŚĆ I m_b-WYJŚĆ

A. Rozpatrujemy podziały zespołu funkcji takie, że $\overline{\overline{X}}_1 \leqslant n_b$
B. Tworzymy tablicę podziałów zespołu funkcji
C. Na podstawie wierszy tablicy podziałów szukamy dekompozycji wielokrotnej maksymalnie ograniczającej liczbę wejść bloku wolnego tzn. $\alpha_{DW} + \overline{\overline{X}}_2 = \min$
D. W przypadku występowania różnych dekompozycji wielokrotnych wybieramy rozwiązanie, dla którego $\alpha_{DW} = \min$
E. Jeżeli po dekompozycji wielokrotnej zbiór $\overline{\overline{X}}_2 \neq \phi$ to poszukujemy dekompozycji iteracyjnej rozpatrując podziały, dla których $X_1 = X_{1i} + X'_2$, gdzie $X'_2 \subset X_2$ (X_{1i} — zbiory związane podziałów tworzących dekompozycję wielokrotną)
F. Jeżeli dekompozycja iteracyjna nie istnieje to identyczną metodą poszukujemy podziałów zbioru $\{g_j(X_{1i}), X_2\}$, $j = 0, 1, \ldots, p-1$
G. W przypadku, gdy nie istnieje dekompozycja ograniczająca liczbę wejść bloku wolnego to dokonujemy prostego podziału wyjść na bloki o liczbie wyjść $m_i \leqslant m_b$ i ponownie poszukujemy podziału „mniejszych" zespołów funkcji p.A

Przykład 5:

Zespół trzech funkcji opisany w zbiorze testowym (ang. benchmark) rd73 podlega między innymi dekompozycjom odpowiadającym poszczególnym wierszom tablicy podziałów przedstawionym na rys. 9. W wybranym fragmencie istnieją dwie pary wierszy rozłącznych (P_i, P_k), (P_j, P_k) stąd, zgodnie z punktem C strategii poszukiwania dekompozycji zespołu funkcji, istnieją dwa rozwiązania. Zgodnie z punktem D wybieramy drugą parę rozłącznych podziałów, prowadzącą do realizacji zbioru testowego przedstawionej na rys. 9.



Rys. 9. Dekompozycja wielokrotna zbioru testowego rd 73

5. WYNIKI EKSPERYMENTÓW

Przedstawiony algorytm dekompozycji został zaimplementowany w systemie Decomp, przeznaczonym między innymi do syntezy układów cyfrowych w strukturach FPGA typu tablicowego. Moduł programowy przeznaczony do dekompozycji umożliwia pracę w trybie interakcyjnym i automatycznym opracowanym dla układów FPGA firmy XILINX serii 3000. W trybie automatycznym zaimplementowany jest algorytm przedstawiony w p. 4.

W trybie interakcyjnym możliwe jest wykonywanie kolejnych dekompozycji. W pierwszym kroku wyznaczana jest tablica podziałów zawierająca podziały, dla których moc zbioru związanego jest mniejsza lub równa od liczby wejść wybranego tablicowego bloku logicznego. Na podstawie wyznaczonych parametrów α i β wybierana jest odpowiednia dekompozycja funkcji i następuje automatycznie przejście (jeśli jest to konieczne) do kolejnego kroku, związanego z analizą bloku wolnego. W sytuacji, gdy istnieje dekompozycja spełniająca warunki $\bar{X}_1 > p$ i $\bar{X}_2 + p < I$ (p — liczba funkcji wiążących; liczba wyjść bloku związanego) możliwe jest wykonanie prostego podziału wyjść [18], stwarzającego „lepsze warunki" do dalszego poszukiwania dekompozycji.

W celu zweryfikowania efektywności przedstawionej metody poszukiwania dekompozycji zespołu funkcji, umożliwiającej jego realizację na blokach logicznych zawierających n_b-wejść i m_b-wyjść, przeprowadzono syntezę układów testowych dla struktur FPGA serii 3000 składających się z konfigurowalnych bloków logicznych (CLB — Configurable Logic Block), zawierających 5-we, 1-wy lub 4-we, 2-wy.

Uzyskane wyniki porównano z wynikami uzyskanymi za pomocą innych algorytmów dekompozycji opublikowanych w pracach [1, 4, 7, 9, 11, 19, 21, 22, 24, 25, 26, 29, 34] (spośród różnych publikacji wybrano najlepsze rozwiązanie). Głównym celem optymalizacji jest zwykle minimalizacja powierzchni lub minimalizacja opóźnień układu po dekompozycji. W tej sytuacji porównano wyniki dekompozycji uzyskane przez system Decomp z wynikami uzyskanymi za pomocą algorytmów służących do minimalizacji powierzchni (liczby wykorzystywanych bloków logicznych — tabl. 1) oraz minimalizacji opóźnień (liczby poziomów logicznych — tabl. 2).

Złożoność algorytmów sprawia, że mogą one być stosowane tylko dla funkcji z niewielką liczbą argumentów (praktycznie kilkunastu). W przypadku wybranych

Tablica 1

Wyniki eksperymentów – minimalizacja powierzchni
(wartości zawarte w tablicy oznaczają liczbę wykorzystywanych bloków)

	ASYL	Chortle	Demain	FGSyn	Mispga	TOS (ISODec)	TRADE	Decomp
5×p1	13	20	9	9	17	–	11	11
9sym	8	41	5	7	7	8	6	6
Bw	27	–		27	27	–	27	27
Clip	33	–	16	18	23	22	29	18
F51m	14	–	10	8	11	–	9	10
Misex1	13	14	8	8	9	–	14	11
Rd73	8	–	5	5	7	7	5	5
Rd84	14	41	7	8	12	10	8	8
Root	–	–	16	–		–	21	19
Sao2	30	–	18	25	28	22	27	22
Z4	4	3	4	4	4	4	4	4

Tablica 2

Wyniki eksperymentów – minimalizacja opóźnień
(pierwsza wartość oznacza liczbę wykorzystywanych bloków, druga – liczbę poziomów logicznych)

	ASYL	Chortle	Demain	FlowMap	Mispga	TRADE	Decomp
5×p1	13/3	21/3	9/2	22/3	21/2	11/2	11/2
9sym	8/3	53/5	5/3	60/5	7/3	6/3	6/3
Bw	27/1	–	–	–	28/1	27/1	27/1
Clip	33/6	83/4	16/2	–	54/4	29/4	18/3
F51m	16/3	–	10/2	–	23/4	9/3	10/3
Misex1	13/2	14	8/2	15/2	–	14/2	11/3
Rd73	8/2	–	5/2	–	8/2	5/2	5/2
Rd84	14/3	41/7	7/2	43/4	13/3	8/3	8/2
Root	–	–	16/3	–	–	21/3	19/3
Sao2	36/3	–	18/3	–	45/5	27/3	22/5
Z4	4/2	4/2	4/2	–	4/2	4/2	4/2

układów testowych sumaryczny czas analizy wszystkich układów wynosi około 15 sekund (PC Pentium Celeron 300 MHz). Maksymalny czas analizy dla układu sao2 wynosi około 9 s. Czas analizy większości wybranych układów testowych jest krótszy od 1 sekundy (rd84,9sym).

6. PODSUMOWANIE

Kluczowym problemem syntezy logicznej układów cyfrowych w oparciu o struktury FPGA typu tablicowego jest problemem podziału układu na poszczególne konfigurowalne bloki logiczne.

Istota przedstawionej w niniejszym artykule metody poszukiwania odpowiedniej dekompozycji złożonej (wielokrotnej lub iteracyjnej) zespołu funkcji polega na analizie wierszy tablicy podziałów. Na podstawie współczynników β_{DW} lub β_{DI} można poszukiwać realizację funkcji logicznej w oparciu o konfigurowalne bloki logiczne o zadanej liczbie wejść. Prosty podział wyjść umożliwia z kolei realizację uzyskanych podukładów na blokach logicznych o zadanej liczbie wyjść.

Niewątpliwą zaletą przedstawionej metody jest możliwość zautomatyzowania całego procesu syntezy. Dalsze prace prowadzone będą w kilku kierunkach tzn. wykorzystania dekompozycji nierozłącznych, optymalizacji procesu zautomatyzowania całej dekompozycji prowadzącej do uzyskiwania układów zawierających minimalną liczbę konfigurowalnych bloków lub minimalną liczbę poziomów logicznych. Pierwsze doświadczenia wskazują na możliwość optymalizacji dekompozycji (minimalizacja liczby bloków/minimalizacja liczby poziomów logicznych) poprzez odpowiednie balansowanie pomiędzy wyborem dekompozycji wielokrotnej lub iteracyjnej oraz podział zespołu funkcji na grupy, które można „lepiej" dekomponować.

BIBLIOGRAFIA

1. R. Abouzeid, B. Babba, M. Crastes, G. Saucier: *Input-Driven Partitioning Methods and Application to Synthesis on Table-Lookup-based FPGAs.* IEEE Trans on CAD, 12, No 7, pp. 913–925, 1993
2. Altera, AMD, Lattice, Xilinx Data Book.
3. R. L. Ashenhurst: *The deecomposition of switching functions.* Proceeding of an International Symposium on the Theory Switching, April 1957
4. B. Babba, M. Crastes, G. Saucier: *Input driven synthesis on PLDs and PGAs.* The European Conference on Desing Automation, Brussels (Belgium), March 1992
5. R. K. Brayton, G. D. Hachtel, C. McMullen and A. L. Sangiovanni-Vincentelli: *Logic Minimization Algorithms for VLSI Synthesis.* Boston, Kluwer Academic Publishers, 1984
6. R. K. Brayton, G. D. Hachtel, A. L. Sangiovanni-Vincentelli: *Multilevel logic synthesis. Proc. of the IEEE,* 78(2), pp. 264–300, 1990
7. S. D. Brown, R. J. Francis, J. Rose, Z. G. Vranesic: *Field Programmable Gate Arrays.* Boston/London/Dotdrecht, Kluwer Academic Publisher, 1993, p. 45–86
8. J. A. Brzozowski, T. Łuba: *Decomposition of Boolean Functions Specified by Cubes.* University of Waterloo Computer Science Departament, CS-97-01, January 1997

9. S. Chang, M. Marek-Sadowska, T. Hwange: *Technology Mapping for TLU FPGSA's Based on Decomposition of Binary Decision Diagrams.* IEEE Transactions on Computer-Aided Desing, 15(10), pp. 1226–1235, October 1996
10. M. J. Ciesielski, S. Yang: *PLADE: A two-stage PLA decomposition.* IEEE Transaction on Computer-Aided Desing, 11(8), pp. 943–954, August, 1992
11. J. Cong, Y. Ding: *FlowMap: An Optimal Technoloigy Mapping Algorithm for Delay Optimization in Lookup-Table Based FPGA Desing.* IEEE Transactions on Computer-Aided Desing, 13(1), pp. 1–12, January 1994
12. H. A. Curtis: *Generalized treee circuit — the basic building of an extended decomposition theory.* J.ACM, vol. 10, pp. 562–581, 1963
13. H. A. Curtis: *The Desing of switching Circuits.* New Jersey, Toronto, New York, D. van Nostrand Company, Inc., Princeton, 1962
14. D. Kania: *Two-lovel synthesis on PALs.* Electronics Letters, Vol. 35, No 11, pp. 879–880, May, 1999
15. D. Kania: *Two-level logis synthesis on PAL-bnased CPLD and FPGA using decomposition.* Euromicro, Milan, Vol. 1, pp. 278–281, 1999
16. D. Kania: *Coding Capacity of Programmable Transcoder.* Electronics and Telecommunications Quarterly, 1998, 44, V. 2, pp. 193–204
17. D. Kania: *Complex decomposition of multiple-output functions.* International Conference on Programmable Devices and Systems, PDS'96, Ostrava, November 26–28, 1996, Czech Republic
18. D. Kania: *Algorytmy podziału wyjść umożliwiające realizację układów cyfrowych w strukturach PLD.* Kwartalnik Elektroniki i Telekomunikacji, 1999, 45, z. 2, ss. 189–202
19. Y. Lai, K. R. Pan, M. Pedram: *OBDD-Based Function Decomposition: Algorithms and Implementation.* IEEE Transactions on Computer-Aided Desing, 15(8), pp. 977–990, August 1996
20. Y. Lai, M. Pedram, S. Vrudhula: *EVBDD-Based Algotithms for Integer Linear Programming Spectral Transformation, and Function Decomposition.* IEEE Transactions on Computer-Aided Desing, 13(8), pp. 959–975, August 1994
21. Ch. Legl, B. Wurth, K. Eckl: *An Implicit Algorithm for Support Minimization during Functional Decomposition.* ED&TC, Paris, pp. 412–417, 1995
22. T. Łuba: *Rola i znaczenie syntezy logicznej w projektowaniu układów cyfrowych.* RUC'99, ss. 75–84, 1999
23. G. Micheli: *Synhesis and optimization of digital circuits.* McGrew-Hill, Inc., 1994
24. R. Murgai, N. Shenoy, R. K. Brayton, A. Sangiovanni-Vincentelli: *Improved Logic Synthesis Algorithms for Table Look up Archtectures.* ICCAD-91, Santa Clara, CA pp. 564–567, November 1991
25. M. Nowicka, T. Łuba, H. Selvaraj: *Multilevel Decomposition Strategies in Decomposition-Based Algorithms and Tools.* International Workshop on Logic and Architecture Synthesis, Grenoble, pp. 129–136, 1997
26. R. Murgai, Y. Nishzaki, N. Shenay, R. K. Brayton, A. Sangiovanni-Vincentelli: *Logic Synthesis for Programmable Gate Array* Proc. 27th DAC, June, pp. 620–625.
27. K. R. Pan, M. Pedram: *FPGA Synthesis for Minimum Area, Delay and Power.* Paris, ED&TC, p. 603, 1996
28. M. Perkowski, R. Malvi, S. Grygiel, M. Burns, A. Mischenko: *Graph Coloring Algorithms for Fast Evaluation of Curtis Decompositions.* 36-th ACM/IEEE DAC'99, New Orleans, June, 1999
29. M. Rawski, M. Nowicka, P. Tomaszewicz, T. Łuba: *Decomposition — based synthesis and its application in FPGA — oriented technology mapping.* PDS'96, pp. 47–54, 1996
30. T. Sasao: *FPGA Desing by Generalized Functional Decomposition in Logic Synthesis and Optimization.* Boston/London/Dotdrecht, Kluwer Academic Publishers, 1993
31. H. Selveraj, T. Łuba, M. Nowicka, B. Bignall: *Multiple-valued decomposition and its applictions in data compresssion and technology mapping.* ICCIMA'97
32. K. Sharma: *Programmable Logic Handbook PLDs, CPLDs, & FPGAs.* McGraw-Hill, 1998

33. F. V o l f: *A bottom-up approach to multiple-level synthesis for look-up table based FPGAs*. Eindhoven, CIT-Data Library 1997
34. W. W a n, M. A. P e r k o w s k i: *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*. Proc. of EDAC'92, pp. 230–235, 1992

D. KANIA

A HEURISTIC DECOMPOSITION METHOD FOR LOOKUP TABLE-BASED FPGA USING COMPLEX DECOMPOSITION OF MULTIPLE-OUTPUT BOOLEAN FUNCTIONS

S u m m a r y

The paper presents the method of decomposition based on the selection of complex decomposition of multiple-output Boolean function. Fundamental to the formulation of such method is Curtis decomposition theory. The constrained algorithm based on a study of row structure of partition table. The algorithm has been implemented in an experimental logic decomposer, Decomp. Experimental results in comparision to previously published methods are given to show the efficiency of the approach.

Key words: logic synthesis, decomposition, partitioning, FPGA

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2000, 46, z. 2, ss. 211–223

New high performance algorithms for sparse matrix envelope compression

ANNA KAMIŃSKA, BEATA PAŃCZYK, MAREK M. STABROWSKI

Katedra Informatyki, Politechnika Lubelska

Otrzymano 2000.02.20
Autoryzowano 2000.04.28

The first of new methods defines the row weights and performs quick sorting of the rows and columns using weights criterion. The second method relies on swapping with diminishing threshold in order to find true global optimum. The third method is a completely redeveloped and modified simulated annealing method. All three methods are compared with two classic algorithms. Matrices used in the tests are more exacting as in other tests up to date.

Key words: banded linear equations, envelope compression, sparse matrices

1. INTRODUCTION

Sparse matrices arise in many areas of science and technique. Some selected examples include the analysis of electrical and electronic circuits, power grids, the design of electrical and magnetic fields, the construction design in mechanical and civil engineering and many others. In most cases sparse matrix of linear equations system has some rudimentary traces of banded form. The banded form reflects the fact that in real world usually the close neighbours interact. In electrical circuit it is equivalent to the observation that only limited number of elements (branches) is connected with a given node. Appropriate numbering of the nodes may enhance the banded features, i.e. compress the nonzero band of the system matrix.

Compression of the nonzero band (condensing nonzeros around main diagonal) is achieved through renumbering of the matrix rows and columns, equivalent to swapping of the rows and columns. Even in present situation of falling prices of computer memories it is advisable to reduce memory requirements, as also processing time is reduced and numerical stability is enhanced. Banded matrix compression is a quite specific form of optimisation or rather minimisation. First of all, it is

performed on multidimensional hypersurface with the dimension equal to the matrix dimension. Next, during this minimisation, the swapping of rows and columns changes the positions of local and global minimum. Thus it is in fact the hunting for a moving target. It is worth to note that for a matrix order n, there exist

$$N_{ord}=n! \tag{1}$$

different permuted matrices. Even for medium size problems with n equal for example to 10^3 the number N_{ord} is exorbitant and brute force approach (exhaustive search) is out of question.

The compression research has a long history but till today it is an open issue. The compression task may be splitted into two subtasks. The first one is the bandwidth compression and the second one — the-envelope or profile compression. These two approaches are closely connected with two different schemes of nonzero band storage. This paper will be confined to the problem of envelope compression.

2. BASIC PARAMETERS OF ENVELOPE COMPRESSION

The intuitive notion of compressed banded matrix should be made more exact through introduction of some quantitative parameters. In the specific case of symmetric matrix A of the order n with elements a_{ij}, the column index of first nonzero element in row i is

$$f_i(A)=\min\{j: a_{ij}\neq 0\} \tag{2}$$

The bandwidth value in a row no i for symmetric matrix is formally defined [5] as

$$\beta_i(A)=i-f_i(A) \tag{3}$$

Envelope (profile) of nonzeros may be further defined [5] as

$$\mathrm{Env}(A)=\{\{i,j\}: 0<i-j\leqslant\beta_i(A)\} \tag{4}$$

More common and more representative is the integral definition of the envelope

$$|\ \mathrm{Env}(A)\ | =\sum_{i=1}^{n}\beta_i(A) \tag{5}$$

This type of definition will be used in the further course. For more general case of unsymmetrical matrix the bandwidth value for row i defined in (3) is rather a left semibandwidth. The right semibandwidth is then calculated with formula

$$\beta_i'(A)=f_i'(A)-i \tag{6}$$

where

$$f_i'(a)=\max\{j: a_{ij}\neq 0\} \tag{7}$$

Total integral envelope for general case of unsymmetrical matrix may be calculated with following formula

$$| \text{Env}(A) | = \sum_{i=1}^{n} (\beta_i(A) + \beta_i'(A)) \tag{8}$$

This definition of envelope will be used in compression algorithms.

3. CLASSIC ENVELOPE COMPRESSION METHODS

Envelope compression methods may be divided into two groups. The methods of the first group perform row and column interchange monitoring the envelope value (8). In some sense they resemble regular optimisation methods with explicit usage of optimum criterion. The methods of the second group do not rely on direct minimisation of envelope value (8). The matrix is transformed according to the algorithms ignoring envelope value. One may suspect that sometimes these algorithms may lead to expansion of the envelope, instead of compression. The results of numerical experiments reported latter will justify this suspicion in some cases.

The first classic compression method [1], belonging to the first group, has been developed by Akyuz and Utku (AU). The algorithm of this method may be summed up as follows:

AU1. Set initial values of top row $i_t = 1$ and bottom row $i_b = n$.

AU2. Try to swap rows and columns i_t and $i_t + 1$. Accept the swapping, if envelope (8) is reduced.

AU3. Try to swap rows and columns i_b and $i_b - 1$. Accept the swapping, if envelope (8) is reduced.

AU4. Increment top pointer i_t by 1 and decrement bottom pointer i_b by 1.

AU5. If $i_t < i_b$ go to step AU2. Else go to step AU1 beginning new series, if there were some accepted swaps in last series or if the number of series is lower than some prescribed limit.

The AU method applies the envelope criterion directly and therefore the envelope cannot increase.

Another, most popular compression method — the reverse Cuthill-McKee (RCM) method [3] – avoids application of envelope criterion. It is a one-pass method, distinctly faster than AU algorithm. Usually the RCM method is formulated in the terms of graph transformations. The graph involved is the graph describing the nonzero structure of a matrix. Non-graph formulation of this method may have following form:

RCM1. Start with an i-th initial matrix row.

RCM2. Append to this row all interconnected rows, i.e. the rows with nonzero elements in column no. i. These rows should be ordered according to decreasing degree, i.e. the number of nonzero entries.

RCM3. Make first row of appended series the subsequent starting row and go to step RCM1 if not all rows have been exhausted.

RCM4. Reverse the ordering of rows/columns obtained in steps RCM1-RCM3 and perform rows and columns permutations.

This method is extremely fast. However it was already mentioned that in some cases no compression but rather band expansion might result from its application. The efficiency of RCM method depends on the selection of initial row/node [5, 6] and the so-called pseudoperiherial node is usually the best.

Other classic compression methods developed by Rosen or by Tewarson are described quite comprehensively elsewhere [8, 9, 10].

4. NEW ENVELOPE COMPRESSION METHODS

Two original envelope compression methods and one thoroughly modified will be presented here.

The first of these methods belongs to the group avoiding the usage of envelope minimisation criterion. It is based on sorting of the rows and columns corresponding to the weights of the rows. The row weight is defined with the formula

$$w_i = \frac{1}{n_{ni}} \sum_{i=1}^{n} (j : a_{ij} \neq 0) \tag{9}$$

where n_{ni} is equal to the number of nonzero entries in the i-th row. The sorting method used here is the efficient quicksort algorithm [4] and therefore the nickname QS for whole method. The QS algorithm runs along following lines:

QS1. Start sorting cycle at first matrix row.

QS2. Compute mean value of all row weights (9) and adopt it as dividing quasi-median for whole rows set.

QS3. Sort the rows/columns into two sets, divided by quasi-median, using general quicksort principle.

QS4. Repeat steps QS2 and QS3 for every subset of the rows until decremented subset size reaches 1.

QS5. Go to step QS1 and repeat whole sorting procedure if any row/column interchanges have been performed in steps QS3 and QS4.

The QS method speed is comparable with that of RCM method. It also shares with RCM method, the lack of robustness, i.e. there is no warranty that the envelope will be really compressed in all cases.

Another new method — thresholded swapping (TS) — applies envelope minimisation criterion directly. It resembles some features of the QS method but there are also distinct differences. One cycle of TS algorithm may be summed up in a bit simplified way as follows:

TS1. Adopt threshold value $t_h (>1)$.

TS2. Start for the sake of simplicity in the first row of a matrix (= current row) and select as candidate for swapping the last n-th row.

TS3. Try to swap current row with the candidate row.

TS4. If trial swapping leads to decrementing of envelope by t_h, accept swapping. Make current row equal to candidate-1 and candidate row equal to current+1. Else decrement candidate row by 1.

TS5. Go to step TS3 if current row index is not equal to candidate row index. Else decrement threshold value t_h by 1 and repeat the steps starting with TS2 until $t_h = 1$.

This method is quite sensitive to the selection of starting point, i.e. current row index in step TS2. It is advisable to start at several other starting points, processing a matrix as a cyclic set of rows and columns. The TS method seems to be at the first sight very time-consuming due to repeated envelope computations. The numerical experiments reported in further course confirm this preliminary conclusion.

Most modern approach to matrix bandwidth and envelope compression uses the ideas of simulated annealing [7]. It combines randomised decisions with rows/columns swapping. Unlike AU method, the pairs of candidates for eventual swap are selected in a purely random way. Next in every stage of simulated annealing (SA) compression, some number of swaps increasing the envelope is accepted. In subsequent stages of SA algorithms, i.e. in lower „temperatures", the number of accepted diverging swaps is steadily decreased [7]. It is worth to note that this seemingly irrational acceptance of divergence from minimum of envelope is based on experimental observations of „moving target" phenomenon during matrix compression.

In our investigation, as the function of energy for optimisation process, we have chosen the envelope, defined by (8) and the annealing parameters as follow:

$T_0 = n*n$ — initial temperature (maximum value of the energy function);

$T_K = 0.9 T_{K-1}$ — annealing schedule for temperature decreasing;

$x_0 = A$ — initial solution;

$K_{max} = n*n$ — maximum number of iteration;

$k = 50$ — specified number of times (number of iteration for constant temperature T_K);

According to above, the SA algorithm proceeds as follow:

SA1. Set initial parameters for simulated annealing schedule. Calculate the energy function for initial solution $F(A)$

SA2. Exchange two randomly chosen columns of matrix A (and corresponding rows if we are dealing with symmetric matrix).

SA3. Calculate energy function for new solution $F(A')$ and $\delta F = F(A') - F(A)$. If $\delta F < 0$ or $e^{\frac{\delta F}{T}} < \text{random}(0,1)$ then set $A = A'$.

SA4. Repeat steps SA2 and SA3 the specified number of times.

SA5. Decrease the temperature according to annealing scheme.

SA6. Repeat steps SA3, SA4, SA5 until we can't find the lower matrix envelope.

SA method efficiency depends on the nature of the problem, i.e. sparsity pattern of the matrix [7]. In some very improbable cases, simulated annealing may lead to band expansion and envelope growth.

5. TEST MATRICES

Three options of test matrix generation have been considered. The first one consisted of deriving of these matrices from real life applications. Such approach, using e.g. the matrices from the so-called Harwell-Boeing tapes, reported in thorough research paper on SA compression [7], is quite specific. There is no room for controlled experiments with matrices of various initial bandwidth, different scarcity and diversified concentration around main diagonal.

The second option considered, started with regular banded matrix, fully populated, i.e. with no zeros inside the band. Next, these matrices were scrambled in a controlled random way. It was hoped that in this way true optimum would be well defined and well known. However almost all algorithms, tested in this research, restored easily the original matrix structures or slightly different regular structures with identical envelope values. Only Akyuz-Utku (AU) algorithm failed sometimes to reach the initial optimum.

Thus it was decided to generate the test matrices as random structures from the scratch. In this approach there is no known optimum state of compression but the task is really exacting. The scarcity pattern is controlled with the aid of four parameters:

1. WIDTH — This parameter spanning the range from 0 to 0.5, determines the bandwidth (rather semibandwidth) with nonzeros. It is a value related to the matrix order.
2. FILLCOEFF — This parameter determines in a bit loose way the density of nonzeros in whole bandwidth.
3. CONCENTRATION — The nonzeros distance to main diagonal is computed as the bandwidth absolute value multiplied by a random number spanning the range between 0 and 1. This random number may be raised to selected power value, resulting in more or less concentrated scarcity pattern. In the experiments reported in following parts of the paper the exponent values of 2 (less concentrated nonzeros) and 4 (nonzeros concentrated closer to main diagonal) have been used.
4. SLIMNESS — If computed nonzero position coincides with already occupied one, two courses may be followed. For a slim nonzero band, matrix generating algorithm simply does nothing, i.e. no nonzero is added. For non-slim bandwidth, the generating algorithm looks for first unoccupied place outside main diagonal and places there new nonzero.

6. EXPERIMENTAL COMPARISON OF COMPRESSION METHODS

Five compression algorithms have been tested and compared. They include AU algorithm in proprietary implementation, RCM algorithm derived from independent source [4] after minor modifications, new original QS and TS algorithms and finally thoroughly revised and modified SA algorithm. The tests have been carried out on Pentium II running Linux 6.0 RedHat operating system. GNU project gcc compiler was used for software development.

Table 1

Compression results for matrix order = 200, WIDTH = 0.25 and parabolic concentration of nonzeros around main diagonal

FILLCOEFF	0.07	0.07	0.05	0.05	0.02	0.02
SLIM	no	yes	no	yes	no	yes
nonzeros	1286	1148	946	844	560	508
envelope	9920	9901	8547	8429	5442	5402
AU algorithm	8617	8353	7265	6975	4299	3851
RCM algorithm	11808	11288	9786	12492	906	580
QS algorithm	8884	8523	6897	6129	2362	1233
TS algorithm	8294	7941	6437	5681	2492	1435
SA algorithm	8325	7965	6543	5786	2547	1602

Table 2

Compression results for matrix order = 200, WIDTH = 0.25 and doubly parabolic concentration of nonzeros around main diagonal

FILLCOEFF	0.07	0.07	0.05	0.05	0.02	0.02
SLIM	no	yes	no	yes	no	yes
nonzeros	1336	922	962	666	570	426
envelope	7322	7197	5850	5732	3546	3439
AU algorithm	6944	6010	5331	4643	2906	2076
RCM algorithm	9561	8326	10026	61432	956	247
QS algorithm	6945	5869	5161	3262	1738	439
TS algorithm	6711	5397	4850	2992	1835	313
SA algorithm	6685	5459	4881	3165	1759	475

The tests have been divided into three series. The first (tab.1) and second (tab.2) series investigated the influence of nonzeros concentration around main diagonal and the influence of fill-in, i.e. the density of nonzeros. The matrix order was constant throughout these tests and equal to 200. The third series of tests (tab.3) was focussed on influence of matrix size on compression efficiency.

In the first series of tests (tab.1) the nonzeros concentrate around main diagonal in randomly parabolic pattern. The fill-in is decremented along the table. Most striking conclusion following from this table analysis is the poor performance of RCM algorithm. With the exception of extremely sparse matrices, the RCM algorithm is inferior to all competitors. Only in the cases of about 3 nonzeros per row the RCM

Table 3

Envelope compression results for matrices with parameters: WIDTH = 0.25, FILLCOEFF = 0.05.

no. of equations	50	100	200	500	1000
nonzeros	–	278	946	6012	23008
envelope	–	1357	8457	78077	369485
AU algorithm	–	1002	7265	73591	357414
RCM algorithm	–	450	9786	95309	459587
QS algorithm	–	656	6897	75454	363812
TS algorithm	–	573	6437	71322	–
SA algorithm	–	582	6543	71599	–

algorithm compresses the envelope. In less sparsely populated matrices with more than 4 nonzeros per row, the RCM algorithm expands the envelope – in some cases even by 50%.

Modest and simple AU algorithm performs the task of envelope compression in more stable way. Envelope compression (tab.1) is in the range of 10% to 30%. It can be observed that for lower fill-in the compression is more efficient.

Fig. 1. St

The new weighted sorting algorithm QS operates in the cases of higher fill-in similarly to AU algorithm. It is worth to note that despite the lack of explicit optimisation criterion (similarly as in reverse Cuthill-McKee algorithm) the envelope was always compressed in the test cases. For extremely sparse matrices the efficiency of QS algorithm is distinctly better than that of AU algorithm but a bit inferior to RCM algorithm.

The TS algorithm, using threshold swapping, is absolutely stable with very good efficiency. Only in the cases of supersparse matrices it is outperformed by RCM algorithm.

Similar to TS properties have SA algorithm — it seems to be quite stable and efficient (except supersparse matrices) but unlucky time consuming.

In the second series of tests (tab. 2) the nonzeros were more tightly concentrated around main diagonal — in doubly parabolic pattern. It can be observed that compression efficiency of all methods, except RCM, has been a bit improved. In the case of extremely sparse matrix (last column in tab. 2) the TS, QS and SA methods efficiency challenges that of RCM method. Similarly as in previous series of the tests, the RCM method expands the envelope for more than 3 nonzeros per row.

In the third series of tests (tab. 3) the influence of matrix size has been investigated. The nonzero fill-in was in the middle of the range investigated in the first test series. Only for smallest matrix of the order equal to 100, the RCM method leads to envelope compression. In other cases it expands the envelope — most spectacularly for largest test matrix. The TS algorithm seems to be the best and stables performer. However it should be noted that the processing times for TS and SA algorithms are very long.

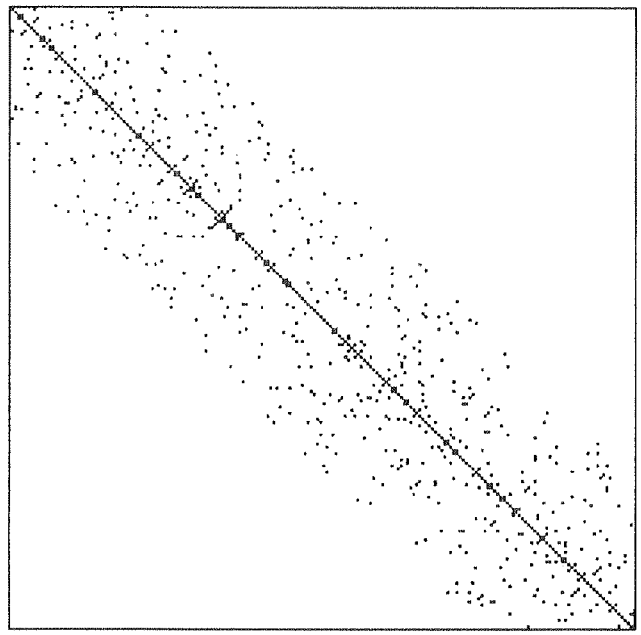


Fig. 1. Structure of original slim matrix for no. of eqs. = 200, WIDTH = 0.25, FILLCOEFF = 0.05 and parabolic concentration.

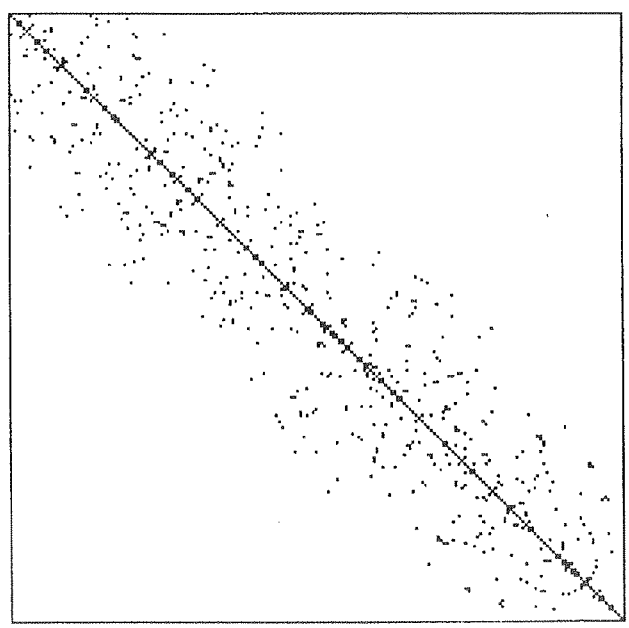


Fig. 2. Structure of the matrix compressed with Akyuz & Utku algorithm.

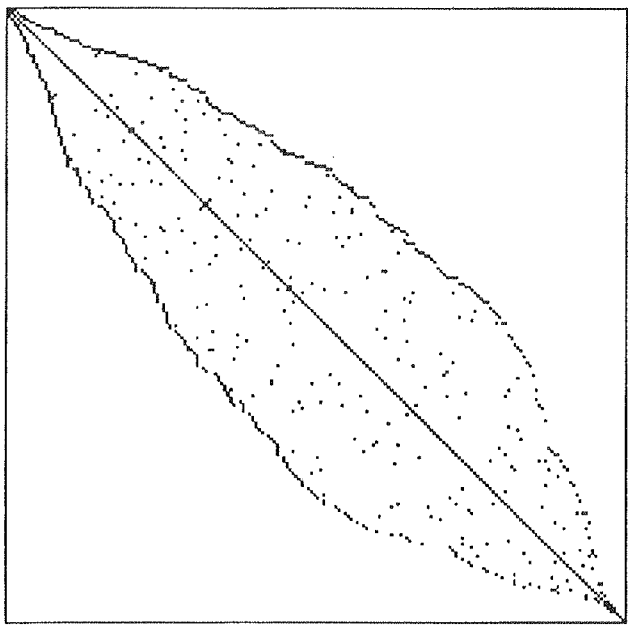


Fig. 3. Structure of the matrix compressed with reverse Cuthill-McKee algorithm.

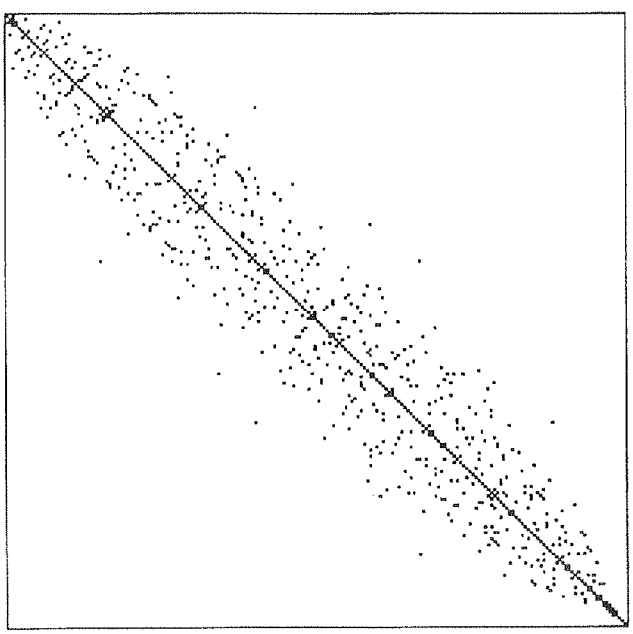


Fig. 4. Structure of the matrix compressed with quicksort algorithm.

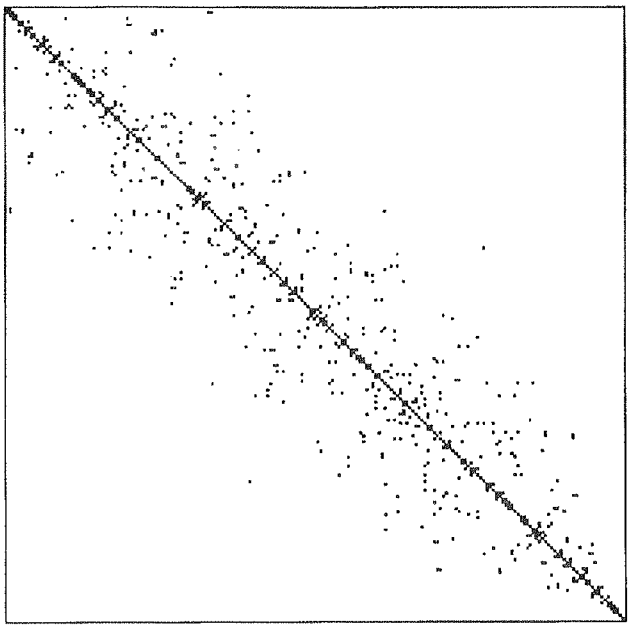


Fig. 5. Structure of the matrix compressed with swapping with threshold algorithm.

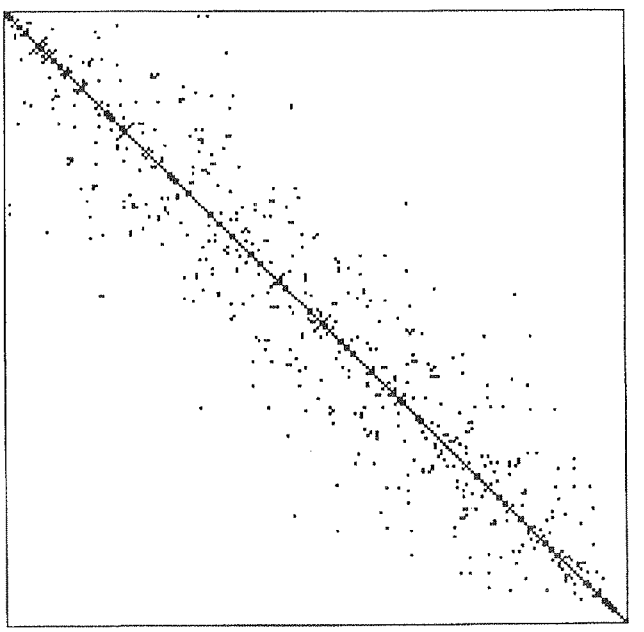


Fig. 6. Structure of the matrix compressed with simulated annealing algorithm.

Poor performance of most popular RCM algorithm can be analysed with the pictures showing population patterns of the matrices. As an example the pictures of the matrices corresponding to the fourth column of tab.1 have been selected and shown (fig. 1 to fig. 6). It should be reminded that fill-in coefficient for original matrix (fig. 1) is equal to 0.05 and the matrix is slim in the sense defined above. It can be observed that all compression methods, except the RCM algorithm (fig. 3) preserve general randomness of the band pattern, introducing some degree of compression. The efficiency of TS (fig. 5) and SA (fig. 6) algorithms is easily observable. In the case of RCM algorithm (fig. 3) quite elegant leaf-like structure of the nonzero band is obtained. However even cursory qualitative inspection of fig. 3 leads to the conclusion that envelope has been vastly enlarged with respect to original matrix (fig. 1).

7. CONCLUSIONS

The envelope compression algorithms have been divided into two groups – with and without explicit minimisation criterion usage. The algorithms with direct usage of minimisation criterion have been found more stable and never leading to nonzero band expansion.

The new algorithms using quick sorting (QS) and swapping with threshold (TS) are more efficient than classic competitive algorithms. Simulated annealing often led to band expansion but envelope is generally decrease. Most popular reverse Cuthill-McKee algorithm is applicable only in the case of small matrices or for extremely sparse ones.

The tests have been carried out in a structured and ordered fashion. The matrix generating algorithm was able to control both fill-in and scarcity pattern, i.e. concentration of nonzeros. Such approach helped to reveal real efficiency of investigated algorithms.

REFERENCES

1. F. A. A k y u z, S. U t k u: *An Automatic Relabelling Scheme for Bandwidth Minimisation of Stiffness Matrices*. AIAA J., 6 (1968), pp. 728-730.
2. T. H. C o r m e n, C. E. L e i s e r s o n, R. L. R i v e s t: *Introduction to Algorithms*. MIT Press, Cambridge Mass., 1994.
3. E. C u t h i l l, J. M c K e e: *Reducing the Bandwidth of Sparse Symmetric Matrices*. Proc. 24th Nat. Conf. ACM Publ. P-69, pp. 157-172. New Jersey, Brandon Systems Press, Princeton, 1968.
4. G. E n g e l n - M u e l l g e s, F. U h l i g: *Numerical Algorithms with C*. Springer Verlag, Berlin 1996.
5. A. G e o r g e, W. H. L i u: *Computer Solution of Large Sparse Positive Definite Systems*. Prentice Hall, Englewood Cliffs 1981.
6. N. E. G i b b s, W. G. P o o l e, P. K. S t o c k m e y e r: *An Algorithm for Reducing the Bandwidth and profile of a Sparse Matrix*. SIAM J. Numer.Anal. 13 (1976), pp. 236-250.
7. R. R. L e w i s: *Simulated Annealing for Profile and Fill Reduction of Sparse Matrices*. Int. J. Num. Meth. Engng., 1993.

8. R. R o s e n: *Matrix Bandwidth Minimisation*. Proc. 23rd Nat. Conf. ACM Publ. P-68, pp. 585-595. Brandon Systems Press, Princeton, New Jersey 1968.
9. R. P. T e w a r s o n: *Sorting and Ordering Sparse Linear Systems' in 'Large Sparse Sets of Linear Equations'* (J.K. Reid ed.), pp. 151-168, Academic Press, New York 1971.
10. R. P. T e w a r s o n: *Sparse Matrices*. Academic Press, New York 1973.

A. KAMIŃSKA, B. PAŃCZYK, M. M. STABROWSKI

NOWE EFEKTYWNE ALGORYTMY KOMPRESJI PROFILU MACIERZY RZADKICH

S t r e s z c z e n i e

Pierwsza z nowych metod definiuje wagi wierszy macierzy oraz wykonuje szybkie sortowanie wierszy i kolumn używając kryterium wagi. Druga metoda polega na przestawianiu wierszy ze zmniejszającą się wartością progową aż do znalezienia minimum globalnego. Trzecia metoda jest całkowicie zmodyfikowaną metodą symulowanego wyżarzania. Wszystkie trzy metody porównano z dwoma algorytmami klasycznymi. Użyte w testach macierze są trudniejsze do skompresowania niż w innych dotychczasowych testach.

Słowa kluczowe: pasmowe układy równań liniowych, kompresja profilu, macierze rzadkie

Eff

expli
sir
These
Utku

Key v

Sparse
circuits, a
most cases
banded fo
only limite

Compr
achieved
swapping
form of m
with the d
the swapp
minimum.

different p
to 10^3 the
question.

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2000, 46, z. 2, ss. 225–235

Efficient bandwidth compression in sparse matrices

ANNA KAMIŃSKA, BEATA PAŃCZYK, MAREK M. STABROWSKI

Katedra Informatyki, Politechnika Lubelska

Otrzymano 2000.02.20
Autoryzowano 2000.04.28

The first of new methods performs quick sorting of the rows and columns without explicit usage of bandwidth criterion. The second swapping method and the third — simulated annealing — rely on direct application of bandwidth optimisation criterion. These methods are compared with classic reverse Cuthill-McKee method and Akyuz and Utku method.

Key words: banded linear equations, bandwidth compression, sparse matrices

1. INTRODUCTION

Sparse matrices arise, among others, in the analysis of electrical and electronic circuits, analysis of power grids, in the design of electrical and magnetic fields. In most cases sparse matrix of linear equations system has some rudimentary traces of banded form. In electrical and electronic circuit it is equivalent to the observation that only limited number of elements (branches) is connected with a given node.

Compression of the nonzero band (condensing nonzeros around main diagonal) is achieved through renumbering of the matrix rows and columns, equivalent to swapping of the rows and columns. Banded matrix compression is a quite specific form of minimisation. First of all, it is performed on multidimensional hypersurface with the dimension equal to the matrix dimension. Next, during this minimisation, the swapping of rows and columns changes the positions of local and global minimum. It is worth to note that for a matrix order n, there exist

$$N_{ord} = n! \tag{1}$$

different permuted matrices. Even for medium size problems with n equal for example to 10^3 the number N_{ord} is exorbitant and naive exhaustive search of minimum is out of question.

The compression research has a long history but till today it is an open issue. The compression task may be splitted into two subtasks. The first one is the bandwidth compression and the second one — the bandwidth or profile compression. These two approaches are closely connected with two different schemes of nonzero band storage. This paper will be devoted to the problem of bandwidth compression.

2. BASIC PARAMETERS OF BANDWIDTH COMPRESSION

The intuitive notion of compressed banded matrix should be made more exact through introduction of some quantitative parameters. In the specific case of symmetric matrix **A** of the order n with elements a_{ij}, the column index of first nonzero element in row i is defined [5] as

$$f_i(A)=\min\{j: a_{ij}\neq 0\} \tag{2}$$

The bandwidth value in a row no i for symmetric matrix is formally defined [5] as

$$\beta_i(A)=i-f_i(A) \tag{3}$$

This type of definition will be used in the further course. For more general case of unsymmetric matrix the bandwidth value for row i defined in (3) is rather a left semibandwidth. The right semibandwidth is then calculated with formula

$$\beta'_i(A)=f'_i(A)-i \tag{4}$$

where

$$f'_i(a)=\max\{j: a_{ij}\neq 0\} \tag{5}$$

The bandwidth for general case of unsymmetric matrix and for whole matrix may be calculated with following formula

$$\beta_{\max}=\max_{i=1}^{n}\beta_i(A)+\max_{i=1}^{n}\beta'_i(A \tag{6}$$

This definition of bandwidth will be used in compression algorithms.

3. CLASSIC BANDWIDTH COMPRESSION METHODS

Bandwidth compression methods may be divided into two groups. The methods of the first group perform row and column interchange controlling the bandwidth value (6). In some sense they resemble regular optimisation methods with explicit usage of optimum criterion. In the methods of the second group the matrix is transformed according to an algorithm ignoring the bandwidth value. These algorithms may lead in some cases to expansion of the bandwidth, instead of compression.

The first classic compression method [1], belonging to the first group, has been developed by Akyuz and Utku (AU). The algorithm of this method may be summed up as follows:

AU1. Set initial values of top row $i_t = 1$ and bottom row $i_b = n$.

AU2. Try to swap rows and columns i_t and $i_t + 1$. Accept the swapping, if bandwidth (6) is reduced.

AU3. Try to swap rows and columns i_b and $i_b - 1$. Accept the swapping, if bandwidth (6) is reduced.

AU4. Increment top pointer i_t by 1 and decrement bottom pointer i_b by 1.

AU5. If $i_t < i_b$ go to step AU2. Else go to step AU1 beginning new series, if there were some accepted swaps in last series or if the number of series is lower than some prescribed limit.

The AU method applies the bandwidth criterion directly and therefore the bandwidth cannot increase.

Another, most popular compression method — the reverse Cuthill-McKee (RCM) method [3] — avoids application of bandwidth criterion. It is a one-pass method, distinctly faster than AU algorithm. Usually the RCM method is formulated in the terms of graph transformations. The graph involved is the graph describing the nonzero structure of a matrix. There is no difference between RCM algorithm in this application and in the envelope minimisation. The description of RCM algorithm may be found elsewhere.

4. NEW BANDWIDTH COMPRESSION METHODS

Two original bandwidth compression methods and one thoroughly modified will be presented here.

The first of these methods belongs to the group avoiding the usage of bandwidth minimisation criterion. It is based on sorting of the rows and columns corresponding to the weights of the rows. The row weight is defined with the formula

$$w_i = \frac{1}{n_{in}} \sum_{i=1}^{n} (j : a_{ij} \neq 0) \tag{7}$$

where n_{ni} is equal to the number of nonzero entries in the i-th row. The sorting method used here is the efficient quicksort algorithm [4] and therefore the nickname QS for whole method. Similarly as in the case of RCM method the description of algorithm will be omitted, as there are no differences with respect to envelope minimisation.

The QS method speed is comparable with that of RCM method. It also shares with RCM method, the lack of robustness, i.e. there is no warranty that the bandwidth will be really compressed in all cases.

Another new method — thresholded swapping (TS) — applies bandwidth minimisation criterion directly. One cycle of TS algorithm may be summed up in a bit simplified way as follows:

TS1. Adopt threshold value $t_h(>1)$.
TS2. Start for the sake of simplicity in the first row of a matrix (= current row) and select as candidate for swapping the last n-th row.
TS3. Try to swap current row with the candidate row.
TS4. If trial swapping leads to decrementing of bandwidth by th, accept swapping. Make current row equal to candidate-1 and candidate row equal to current+1. Else decrement candidate row by 1.
TS5. Go to step TS3 if current row index is not equal to candidate row index. Else decrement threshold value t_h by 1 and repeat the steps starting with TS2 until $t_h=1$.

This method is quite sensitive to the selection of starting point, i.e. current row index in step TS2. It is advisable to start at several other starting points, processing a matrix as a cyclic set of rows and columns. The TS method is very time-consuming due to repeated bandwidth computations.

Most modern approach to matrix bandwidth and bandwidth compression uses the ideas of simulated annealing [7]. It combines randomised decisions with rows/columns swapping. Unlike AU method, the pairs of candidates for eventual swap are selected in a purely random way. Next in every stage of simulated annealing (SA) compression, some number of swaps increasing the bandwidth is accepted. In subsequent stages of SA algorithms, i.e. in lower „temperatures", the number of accepted diverging swaps is steadily decreased [7]. It is worth to note that this seemingly irrational acceptance of divergence from minimum of bandwidth is based on experimental observations of „moving target " phenomenon during matrix compression.

In our investigation, as the function of energy for optimisation process, we have chosen the bandwidth, defined by (8) and the annealing parameters as follow:

$T_0=n*n$ — initial temperature (maximum value of the energy function);
$T_K=0.9T_{K-1}$ — annealing schedule for temperature decreasing;
$x_0=A$ — initial solution;
$K_{max}=n*n$ — maximum number of iteration;
$k=50$ — specified number of times (number of iteration for constant temperature T_K);

According to above, the SA algorithm proceeds as follow:

SA1. Set initial parameters for simulated annealing schedule. Calculate the energy function for initial solution $F(A)$
SA2. Exchange two randomly chosen columns of matrix A (and corresponding rows if we are dealing with symmetric matrix).
SA3. Calculate energy function for new solution $F(A')$ and $\delta F=F(A')-F(A)$. If $\delta F<0$ or $e^{-\frac{\delta F}{T}}<\text{random}(0,1)$ then set $A=A'$.
SA4. Repeat steps SA2 and SA3 the specified number of times.
SA5. Decrease the temperature according to annealing scheme.
SA6. Repeat steps SA3, SA4, SA5 until we can't find the lower matrix bandwidth.

SA method efficiency depends on the nature of the problem, i.e. sparsity pattern of the matrix [7]. In some very improbable cases, simulated annealing may led to band expansion and envelope growth.

5. TEST MATRICES

The test matrices have been generated as random structures starting from the scratch. In this approach there is no known optimum state of compression but the task is really exacting. The sparsity pattern is controlled with the aid of four parameters:

1. WIDTH — This parameter spanning the range from 0 to 0.5, determines the bandwidth (rather semibandwidth) with nonzeros. It is a value related to the matrix order.
2. FILLCOEFF — This parameter determines in a bit loose way the density of nonzeros in whole bandwidth.
3. CONCENTRATION — The nonzeros distance to main diagonal is computed as the bandwidth absolute value multiplied by a random number spanning the range between 0 and 1. This random number may be raised to selected power value, resulting in more or less concentrated sparsity pattern. In the experiments reported in following parts of the paper the exponent values of 2 (less concentrated nonzeros) and 4 (nonzeros concentrated closer to main diagonal) have been used.
4. SLIMNESS — If computed nonzero position coincides with already occupied one, two courses may be followed. For a slim nonzero band, matrix generating algorithm simply does nothing, i.e. no nonzero is added. For non-slim bandwidth, the generating algorithm looks for first unoccupied place outside main diagonal and places there new nonzero.

6. EXPERIMENTAL COMPARISON OF BANDWIDTH COMPRESSION METHODS

Five compression algorithms have been tested and compared. They include AU algorithm in proprietary implementation, RCM algorithm derived from independent source [4] after minor modifications, new original QS and TS algorithms and finally thoroughly revised and modified SA algorithm. The tests have been carried out on Pentium II running Linux 6.0 RedHat operating system. GNU project gcc compiler was used for software development.

The tests have been divided into two series. Both series investigated the influence of nonzeros concentration around main diagonal and the influence of fill-in, i.e. the density of nonzeros. The matrix order was constant throughout these tests and equal to 200. In the first series (tab.1) the concentration of nonzeros around main diagonal was parabolic. In the second series (tab.2) the nonzeros were closer to main diagonal due to doubly parabolic concentration. The fill-in is decremented along the table.

Most general conclusion following from both tables analysis is the observation that bandwidth compression is really difficult task. Usable results can be achieved only in the case of extremely sparse matrices or in the case of higher initial concentration of nonzeros around main diagonal.

Table 1

Bandwidth compression results for matrix order = 200, WIDTH = 0.25 and parabolic concentration of nonzeros around main diagonal

FILLCOEFF	0.07	0.07	0.05	0.05	0.02	0.02
SLIM	no	yes	no	yes	no	yes
nonzeros	1286	1148	946	844	560	508
bandwidth	98	98	98	98	98	98
AU algorithm	98	98	98	98	96	96
RCM algorithm	94	108	88	120	22	12
QS algorithm	106	102	96	104	66	70
TS algorithm	98	98	98	98	94	94
SA algorithm	98	98	98	98	98	98

Table 2

Bandwidth compression results for matrix order = 200, WIDTH = 0.25 and doubly parabolic concentration of nonzeros around main diagonal

FILLCOEFF	0.07	0.07	0.05	0.05	0.02	0.02
SLIM	no	yes	no	yes	no	yes
nonzeros	1336	922	962	666	570	426
bandwidth	98	98	98	98	98	98
AU algorithm	98	98	98	98	94	94
RCM algorithm	78	70	100	74	14	6
QS algorithm	90	94	82	78	48	18
TS algorithm	98	98	98	98	90	86
SA algorithm	98	98	98	98	98	98

Fig. 1. S

It can be observed in tab. 1 that all algorithms, down to the fill-in coefficient of 0.05, do not lead to distinct bandwidth compression. Moreover, the algorithm without explicit application of bandwidth criterion (RCM and QS) simply expand the bandwidth, even by 20% (RCM). The algorithms applying bandwidth criterion only do not spoil the situation. The situation is better in the case of fill-in coefficient equal to 0.2, which is equivalent to less than 3 nonzeros per row. The algorithms without bandwidth criterion, i.e. the classic reverse Cuthill-McKee method and new quicksort algorithm, succeed in squeezing the bandwidth. The achievements of stable algorithms with bandwidth criterion are very modest – below 5% of bandwidth reduction.

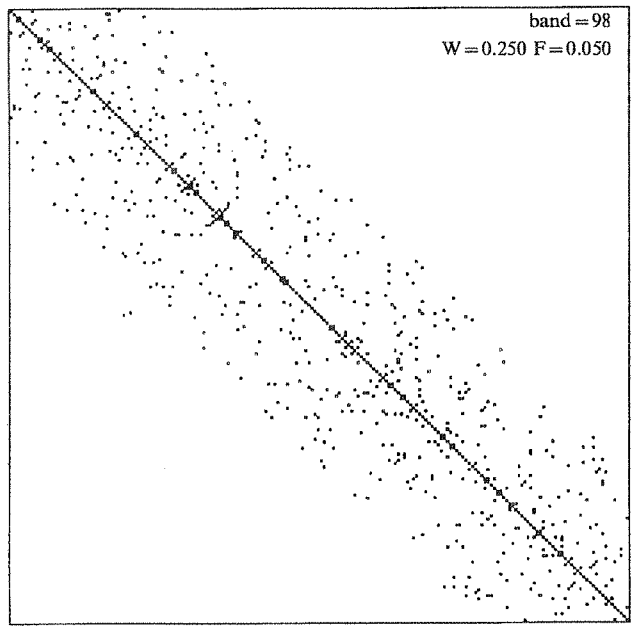


Fig. 1. Structure of original slim matrix for no. of eqs. = 200 WIDTH = 0.25, FILLCOEFF = 0.05 and parabolic concentration.

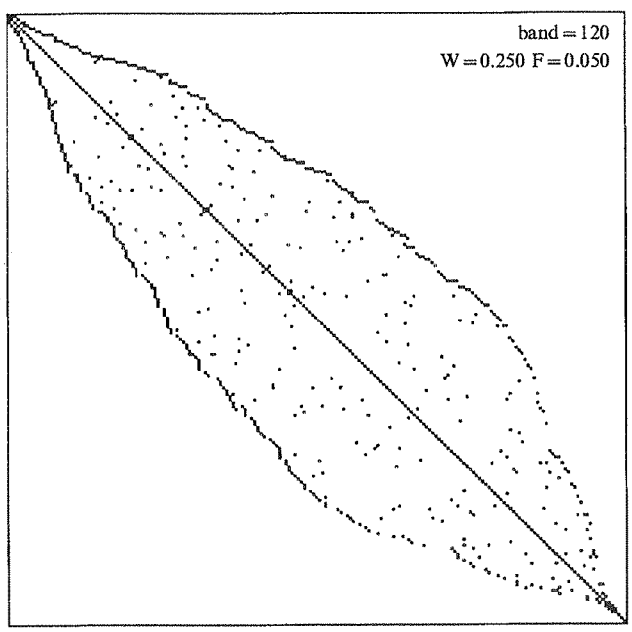


Fig. 2. Structure of the matrix compressed with Akyuz & Utku algorithm.

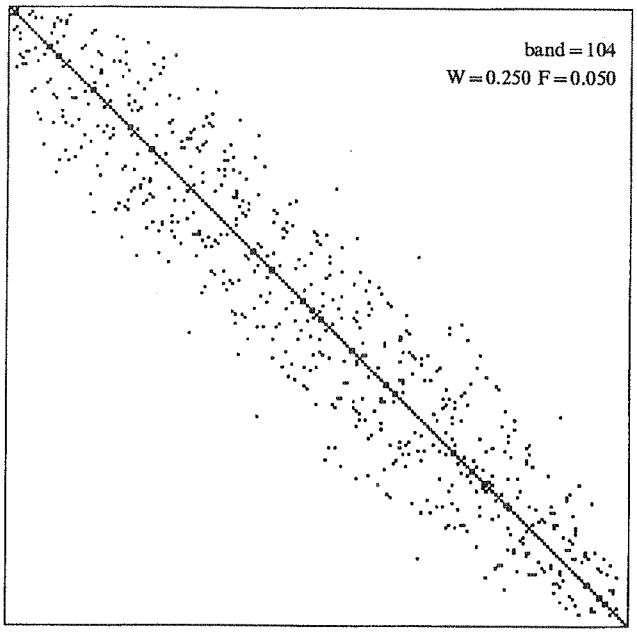


Fig. 3. Structure of the matrix compressed with reverse Cuthill-McKee algorithm.

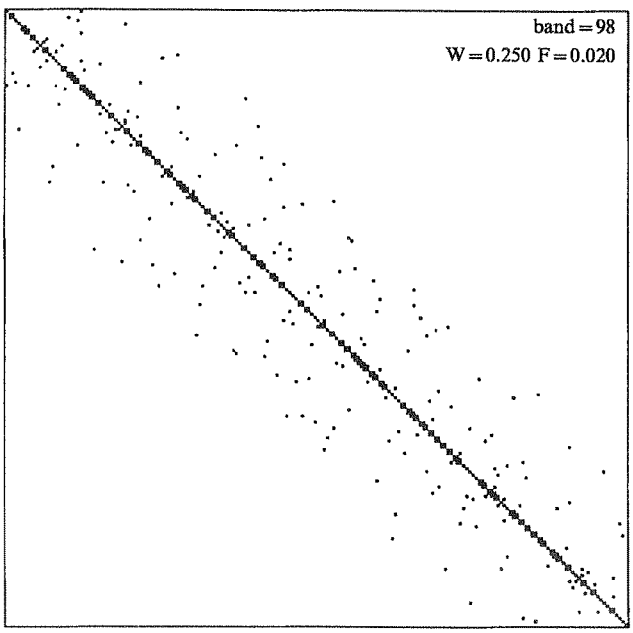


Fig. 4. Structure of the matrix compressed with quicksort algorithm.

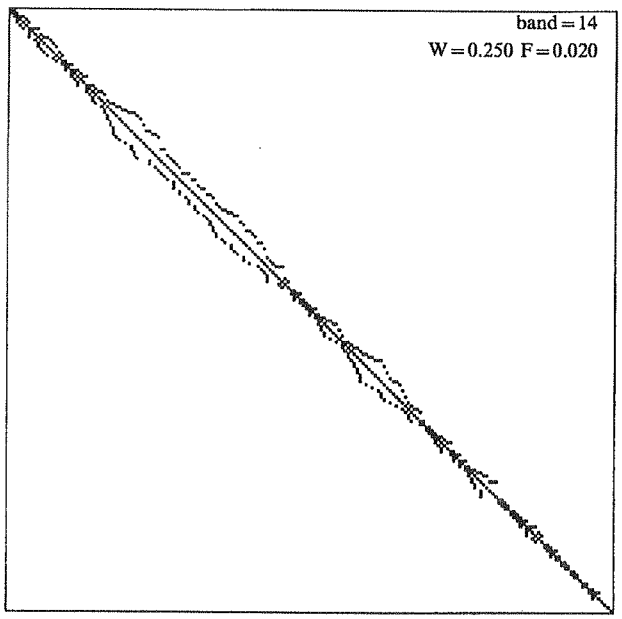


Fig. 5. Structure of the matrix compressed with swapping with threshold algorithm.

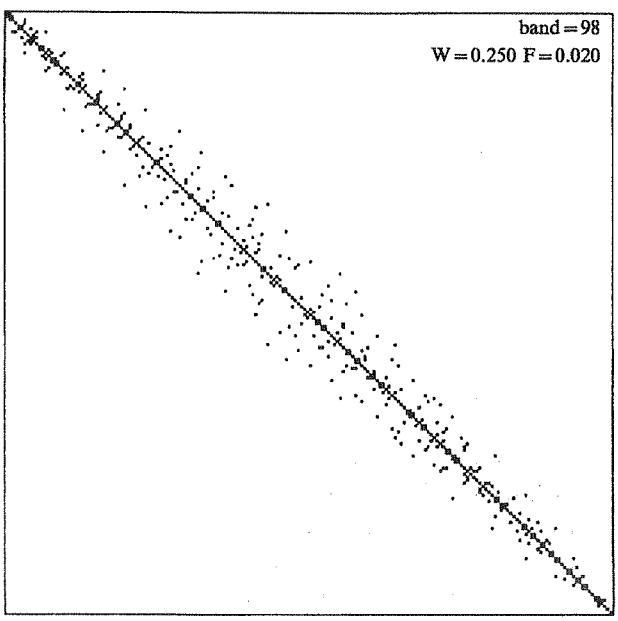


Fig. 6. Structure of the matrix compressed with simulated annealing algorithm.

The results reported in tab. 2 are much better but it can be clearly attributed to doubly parabolic initial concentration of nonzeros. Here only in one exceptional case the classic RCM algorithm leads to expansion of the nonzero band. All other algorithms offer always some degree of bandwidth compression or at least do not spoil the situation. Quite good and stable performer is here the quicksort algorithm. Better results but only in some cases are achieved with the aid of classic RCM algorithm. The algorithms using explicitly minimisation criterion do not shine in this comparison. Only in the case of lowest nonzeros density (fill-in coefficient equal to 0.02) some improvement can be observed.

More insight into the bandwidth compression may be gained through inspection of the pictures showing matrix population patterns. By the way, one of the reasons of limiting matrix size to 200 was the readability of matrix pictures. The first series of matrix pictures corresponds to one column of tab.1 with parabolic nonzeros concentration, fill-in coefficient equal to 0.05 and slim matrix structure. Original matrix (fig. 1) is expanded by both reverse Cuthill-McKee algorithm (fig. 2) and quicksort algorithm (fig. 3). It may be observed that RCM algorithm produces elegant and regular leaf-like structure of processed matrix (fig. 2). The outer nonzeros in adjacent rows are located in neighbouring columns. However it is clearly visible that the bandwidth has been markedly expanded. In clear contrast the QS algorithm (nonzero band has been a bit expanded) leaves the boundaries of transformed nonzero band ragged. Another series of matrix pictures is more optimistic. It corresponds do tab. 2 (higher – doubly parabolic concentration), fill-in of 0.02 and non-slim structure. Original uncompressed matrix (fig. 4) is less densely populated. Due to lower density the results obtained with RCM algorithm (fig. 5) are almost excellent. The leaf-like structure of transformed matrix is difficult to identify as there are less than 3 nonzeros per row. In the case of QS algorithm (fig. 6) the results are inferior to that obtained with RCM algorithm. However the reduction of the bandwidth to half of the initial value are easy to asses.

7. CONCLUSIONS

The bandwidth compression algorithms have been divided into two groups — with and without explicit minimisation criterion usage. The algorithms with direct usage of minimisation criterion have been found more stable (no band expansion) but also less efficient in the cases lending to bandwidth compression.

The new algorithm using quick sorting (QS) leads sometimes to better results than popular reverse Cuthill-McKee algorithm. However both algorithms improve the bandedness of the matrix only in the case of extremely sparse nonzero patterns. Generally the task of bandwidth compression is much more difficult than envelope compression. The results are also less spectacular and profitable.

The tests have been carried out in a structured and ordered fashion. The matrix generating algorithm was able to control both fill-in and sparsity pattern, i.e. concentration of nonzeros. Such approach helped to reveal real efficiency of investigated algorithms.

REFERENCES

1. F. A. A k y u z, S. U t k u: *An Automatic Relabelling Scheme for Bandwidth Minimisation of Stiffness Matrices*. AIAA J., 6 (1968), pp. 728-730.
2. T. H. C o r m e n, C. E. L e i s e r s o n, R. L. R i v e s t: *Introduction to Algorithms'*. MIT Press, Cambridge Mass., 1994.
3. E. C u t h i l l, J. M c K e e: *Reducing the Bandwidth of Sparse Symmetric Matrices*. Proc. 24th Nat. Conf. ACM Publ. P-69, pp. 157-172. New Jersey, Brandon Systems Press, Princeton, 1968.
4. G. E n g e l n - M u e l l g e s, F. U h l i g: *Numerical Algorithms with C*. Berlin, Springer Verlag, 1996.
5. A. G e o r g e, W. H. L i u: *Computer Solution of Large Sparse Positive Definite Systems*. Prentice Hall, Englewood Cliffs 1981.
6. N. E. G i b b s, W. G. P o o l e, P. K. S t o c k m e y e r: *An Algorithm for Reducing the Bandwidth and profile of a Sparse Matrix*. SIAM J. Numer.Anal. 13 (1976), pp. 236-250.
7. R. R. L e w i s: *Simulated Annealing for Profile and Fill Reduction of Sparse Matrices*. Int. J. Num. Meth. Engng., 1993.
8. R. R o s e n: *Matrix Bandwidth Minimisation*. Proc. 23rd Nat. Conf. ACM Publ. P-68, pp. 585-595. Brandon Systems Press, Princeton, New Jersey 1968.
9. R. P. T e w a r s o n: *Sorting and Ordering Sparse Linear Systems in Large Sparse Sets of Linear Equations* (J.K. Reid ed.), pp. 151-168, Academic Press, New York 1971.
10. R. P. T e w a r s o n: *Sparse Matrices*. Academic Press, New York 1973.

A. KAMIŃSKA, B. PAŃCZYK, M. M. STABROWSKI

EFEKTYWNA KOMPRESJA PASMA MACIERZY RZADKICH

S t r e s z c z e n i e

Pierwsza z nowych metod dokonuje szybkiego sortowania wierszy i kolumn bez jawnego użycia kryterium szerokości pasma. Druga — metoda progowa oraz trzecia — symulowanego wyżarzania — polegają na bezpośrednim zastosowaniu kryterium optymalizacji wartości szerokości pasma. Metody te zostały porównane z klasyczną odwrotną metodą Cuthill-Mc-Kee oraz z metodą Akyuza-Utku.

Słowa kluczowe: pasmowe układy równań liniowych, kompresja pasma, macierze rzadkie

W
bierny
zane
metod
zakłóc
towan
dopod
projek
charak
takich

Słowa

Radiol
zamierzon
mentów o
płatki śnie
równomier
roślinność
powstawa
jak [7, 12]:

- zakłóce
- zakłóce
- zakłóce
- zakłóce

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2000, 46, z. 2, ss. 237–253

Metody i algorytmy symulacji radiolokacyjnych zakłóceń biernych

ANDRZEJ JAKUBIAK

Instytut Telekomunikacji, Politechnika Warszawska

Otrzymano 2000.04.04
Autoryzowano 2000.05.04

W artykule przedstawiono metody i algorytmy symulacji radiolokacyjnych zakłóceń biernych. Omówiono modele matematyczne, służące do opisu tych zakłóceń, ściśle powiązane z danymi eksperymentalnymi. Dokonano krótkiej analizy znanych z literatury metod, pozwalających na generowanie niezależnych bądź skorelowanych ciągów próbek zakłóceń niegaussowskich, takich jak logarytmonormalne, Weibulla i typu K. Prezentowane algorytmy umożliwiają pełną kontrolę zarówno rozkładów brzegowych prawdopodobieństwa jak i zależności korelacyjnych, co czyni je szczególnie przydatnymi do projektowania radiolokacyjnych torów odbiorczych. Pomimo to, że niniejszy artykuł ma charakter przeglądowy, to jego oryginalną składową jest wyodrębnienie i szczegółowy opis takich generatorów zakłóceń, które można stosować bezpośrednio w procesach symulacji.

Słowa kluczowe: radiolokacja, zakłócenia bierne, symulacja komputerowa.

1. WSTĘP

Radiolokacyjne zakłócenia bierne powstają w wyniku przypadkowych i niezamierzonych odbić sygnału sondującego od różnego rodzaju naturalnych elementów odbijających. Z reguły są nimi opady atmosferyczne (krople deszczu, płatki śniegu, grudki lodu), chmury, turbulencje powietrza, fale morskie, nierównomierności powierzchni ziemi (góry, wsoka zabudowa), pokrywająca ziemię roślinność (lasy), stada ptaków i inne naturalne przeszkody. Ze względu na źródło powstawania wyróżnia się najczęściej takie podstawowe typy zakłóceń biernych, jak [7, 12]:

- zakłócenia od powierdzchni ziemi,
- zakłócenia meteorologiczne,
- zakłócenia od powierzchni morza,
- zakłócenia od stad ptaków.

Typowym skutkiem działania zakłóceń biernych jest swoisty chaos na wskaźniku radiolokacyjnym, poprzez pojawienie się nieoczekiwanych i trudnych do identyfikacji ech. Fakt ten znajduje odbicie w nazwie anglojęzycznej tego typu zakłóceń — *clutter* — co w mowie potocznej oznacza zamieszanie, bałagan. Niestety, taka sytuacja stwarza bardzo poważne zgrożenie dla bezpieczeństwa nawigacji prowdzonej za pomocą radaru, a nawet przy braku bezpośredniego zagrożenia utrudnia bądź uniemożliwia poprawną identyfikację i detekcję obserwowanych celów. Zatem zachodzi konieczność projektowania torów odbiorczych eliminujących w jak największym stopniu niekorzystne oddziaływanie zakłóceń biernych, co wymaga znajomości charakterystycznych parametrów tychże zakłóceń. Na podstawie parametrów tworzy się modele matematyczne, które opisując w przybliżeniu procesy rzeczywiste umożliwiają prowadzenie odpowiednich symulacji. Modele te, ze względu na losowy charakter zakłóceń, oparte są głównie na procesach stochastycznych o określonych statystykach pierwszego i drugiego rzędu. Duża ilość opublikowanych w ostatnich latach danych pomiarowych [1, 2, 9] umożliwiła skorygowanie panującego jeszcze do lat sześćdziesiątych przekonania, że rozkład prawdopodobieństwa próbek amplitudy zakłóceń ma charakter gaussowski. Obecnie przyjmuje się powszechnie, że zakłócenia bierne są niegaussowskie, co wynika przede wszystkim ze stosowania radarów o coraz większej rozdzielczości i co potwierdzają liczne dane eksperymentalne [1, 2, 3, 8, 9]. Przy takim podejściu podstawowym problemem jest dobór właściwego rozkładu prawdopodobieństwa i jego parametrów. W przypadku statystyk drugiego rzędu badania koncentrują się na określeniu kształtu widma mocy i korelacji przestrzennej zakłóceń [4].

Znajomość opisu matematycznego pozwala na zbudowanie modeli i symulację komputerową zakłóceń biernych o cechach zbliżonych do rzeczywistych. Jakość symulowanego sygnału zależy nie tylko od przyjętego modelu matematycznego, zależy także od zastosowanych procedur generacyjnych. Dlatego tym zagadnieniom, to znaczy modelowaniu matematycznemu oraz metodom generowania próbek radiolokacyjnych zakłóceń biernych, jest poświęcony niniejszy artykuł.

Intencją autora było możliwie pełne przedstawienie algorytmów, pozwalających na symulację zakłóceń biernych o znanych i najczęściej stosowanych modelach probabilistycznych. Wybrano metody umożliwiające generowanie ciągów losowych o zadanych zarówno statystykach pierwszego rzędu jak i właściwościach korelacyjnych. Algorytmy zostały podane w sposób szczegółowy, przydatny bezpośrednio do implementacji w postaci procedur komputerowych.

W pracy można wyróżnić dwie części. W pierwszej zaprezentowano modele matematyczne, używane do opisu radiolokacyjnych zakłóceń biernych. Przytoczono dane eksperymentalne, na podstawie których został określony zakres przydatności omawianych modeli oraz typowe wartości ich parametrów. W części drugiej przedstawiono metody generacji ciągów losowych, wskazując na możliwości zastosowania w symulacji poszczególnych rodzajów zakłóceń. W oddzielnych rozdziałach podano algorytmy dla próbek niezależnych i skorelowanych.

2. MODELE MATEMATYCZNE ZAKŁÓCEŃ

Zakłócenia bierne ze względu na mechanizm ich powstawania mają charakter losowy. W związku z tym muszą być one modelowane procesem stochastycznym o odpowiednich właściwościach probabilistycznych. Zazwyczaj przyjmuje się, że sygnał pasmowy, skupiony wokół pulsacji ω:

$$x(t = \mathrm{Re}[\tilde{X}e^{j\omega t}] \tag{1}$$

gdzie $\tilde{X}$ — zespolona obwiednia losowa, reprezentuje radiolokacyjne zakłócenia bierne [5, 6]. Obwiednię zespoloną można przedstawić w postaci iloczynu:

$$\tilde{X} = Xe^{j\Theta} \tag{2}$$

gdzie X — amplituda losowa, Θ — faza losowa.

Wyznaczenie właściwego opisu probabilistycznego obydwu elementów losowych, to jest amplitudy X i fazy Θ, stanowi jedno z najistotniejszych zadań w procesie modelowania zakłóceń. Pod pojęciem „wyznaczenie właściwego opisu" rozumie się znalezienie takiego, który jak najlepiej odzwierciedla rzeczywiste cechy probabilistyczne odbieranego sygnału zakłócającego i jednocześnie daje się w realny sposób wykorzystać w procedurach symulacyjnych i pracach projektowych. Jednocześnie spełnienie tych dwóch warunków jest na ogół trudne, ponieważ dokładniejszy model wymaga bardziej skomplikowanego opisu matematycznego. Z dostępnej literatury wynika, że pomimo znacznej liczby prezentowanych w ostatnich latach wyników rejestracji próbek zakłóceń nie udało się do dziś stworzyć uniwersalnego modelu probabilistycznego tych zakłóceń. Skłania to do sformułowania wniosku, że ze względu na fizykę zjawiska, różnorodność przyczyn i warunków w jakich powstaje, trudno oczekiwać w jabliższym czasie powstania takiego uniwersalnego modelu.

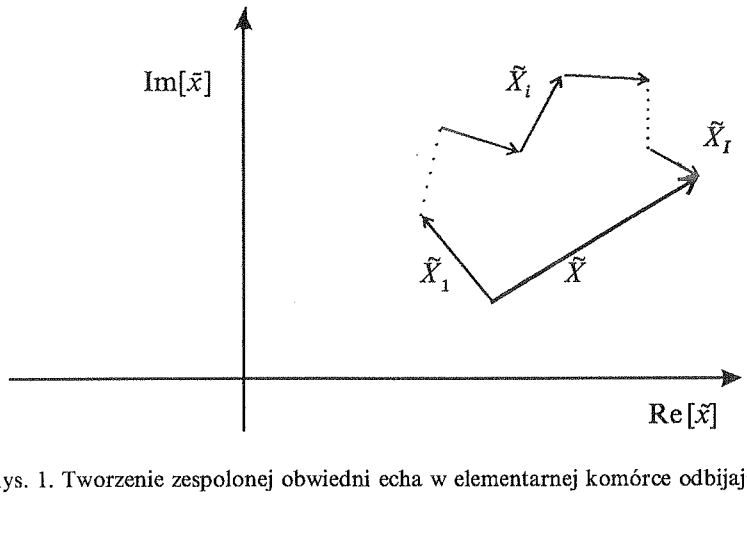
Sygnał odbierany przez rada (nazywany potocznie echem) jest częścią fali elektromagnetycznej, odbitej od obiektów, znajdujących się na drodze jej propagacji. Przyjmuje się powszechnie, że tak zwana elementarna komórka radaru, charakteryzowana przez szerokość kątową listka głównego charakterystyki promieniowania anteny i długość impulsu sondującego, zawiera wiele elementów rozpraszających. Wskutek tego zespolona obwiednia $\tilde{X}$ jest sumą wektorową [5, 6]:

$$\tilde{X} = Xe^{j\Theta} = \sum_{i=1}^{I} \tilde{X}_i = \sum_{i=1}^{I} X_i e^{j\Theta_i} \tag{3}$$

gdzie X_i — losowa amplituda echa od i-tego elementu rozpraszającego, Θ_i — losowa faza echa od i-tego elementu rozpraszającego, I — ilość elementów rozpraszających w komórce elementarnej.

Sumowanie odbić $\tilde{X}_i$ od pojedynczych elementów rozpraszających na płaszczyźnie zespolonej ilustruje rys. 1.

Opisy probabilistyczne amplitudy X i fazy Θ zależą od składowych X_i i Θ_i, które są wzajemnie niezależnymi zmiennymi losowymi. Funkcję gęstości prawdopodobieństwa zmiennej Θ przyjmuje się w postaci:



Rys. 1. Tworzenie zespolonej obwiedni echa w elementarnej komórce odbijającej

$$p(\Theta)=\frac{1}{2\pi},\ 0\leqslant\Theta\leqslant 2\pi, \tag{4}$$

której użyteczność dla praktyki potwierdzono dużą liczbą danych eksperymentalnych [6, 7, 12].

W przypadku funkcji gęstości prawdopodobieństwa amplitudy X sprawa jest bardziej złożona. Do końca lat sześćdziesiątych, kiedy radary cechowała stosunkowo niska rozdzielczość (długi impuls sondujący, szeroka charakterystyka antenowa), przyjmowano **gaussowski model zakłóceń** [7]. Wynikało to z następujących przesłanek:

- ze względu na znaczny rozmiar przestrzenno-czasowy w komórce elementarnej znajduje się duża liczba I niezależnych elementów rozpraszających,
- duża liczba I umożliwia zastosowanie centralnego twierdzenia granicznego przy wyznaczaniu rozkładu prawdopodobieństwa amplitudy X.

Przyjęcie powyższych przesłanek prowadzi do wniosku, że niezależnie od rozkładów prawdopodobieństwa składowych $\tilde{X}_i$, obie składowe ortogonalne obwiednie $\tilde{X}$ mają rozkład gaussowski. Oznacza to, że amplituda losowa X ma funkcję gęstości prawdopodobieństwa opisaną rozkładem Rayleigha [14, 15]:

$$p(X)=\frac{2}{\sigma}\left(\frac{X}{\sigma}\right)\exp\left[-\left(\frac{X}{\sigma}\right)^2\right],\ X\geqslant 0 \tag{5}$$

gdzie σ — parametr skali.

Momenty rozkładu Rayleigha, takie jak wartość średnia m_x i średniokwadratowa $E\{X^2\}$ wynoszą:

$$m_X=\frac{\sqrt{\pi}}{2}\sigma \tag{6}$$

$$E\{X^2\}=\sigma^2 \tag{7}$$

Obserwacja danych eksperymantalnych w wielu przypadkach potwierdzała dobre dopasowanie modelu Rayleigha do zakłóceń rzeczywistych [7]. Wzrost rozdzielczości radarów (węższa wiązka antenowa, krótsze impulsy sondujące) sprawił, że uzyskiwane dane pomiarowe w znacznym stopniu zaczęły odbiegać od tego modelu. Empiryczne rozkłady prawdopodobieństwa miały większy „ogon" (wyższe wartości dla dużych amplitud) oraz większą wartość odchylenia standardowego w stosunku do wartości średniej [3, 8]. Zjawisko to daje się wytłumaczyć między innymi mniejszymi rozmiarami elementarnej komórki odbijającej, na tyle małą liczbą elementów rozpraszających, że nie można stosować centralnego twierdzenia granicznego. Próby dopasowania rozkładów prawdopodobieństwa do danych pomiarowych doprowadziły w efekcie do powstania wielu tak zwanych **niegaussowskich modeli zakłóceń** [3, 6, 10].

W ciągu ostatnich kilku lat pomiary echa od powierzchni lądu, morza, chmur, opadów i stad ptaków pozwoliły na ograniczenie stosowanych modeli niegaussowskich do trzech, a mianowicie **logarytmo-normalnego, Weibulla oraz typu K** [1, 2, 3, 9]. Pierwsze dwa modele powstały w zasadzie jako wynik dopasowywania do danych eksperymentalnych, trzeci — typu K — powstał na podstawie analizy zjawiska tworzenia wypadkowego wektora obwiedni zakłóceń z uwzględnieniem fluktacji czasowo przestrzennej powierzchni odbijającej [5, 10, 11]. Do dziś nie zostało jednoznacznie rozstrzygnięte, który z nich jest najlepszy i najbardziej uniwersalny. Dlatego w niniejszej pracy dalsza analiza zostanie ograniczona do tych trzech wymienionych modeli, jako aktualnie „obowiązujących" i powszechnie stosowanych. Są to modele dwuparametryczne — jeden z parametrów funkcji gęstości prawdopodobieństwa amplitudy jest parametrem skali, drugi kształtu. Parametr kształtu odgrywa zasadniczą rolę w dopasowaniu modelu teoretycznego do danych pomiarowych, natomiast parametr skali można unormować do wartości równej jeden poprzez prosty zabieg, a mianowicie normalizację amplitudy, zgodnie z zależnością:

$$X_{norm} = \frac{X}{m_x} \tag{8}$$

gdzie m_x oznacza wartość średnią zmiennej X.

W **modelu logarytmo-normalnym** (MLN) funkcja gęstości prawdopodobieństwa amplitudy zakłóceń wyraża się wzorem:

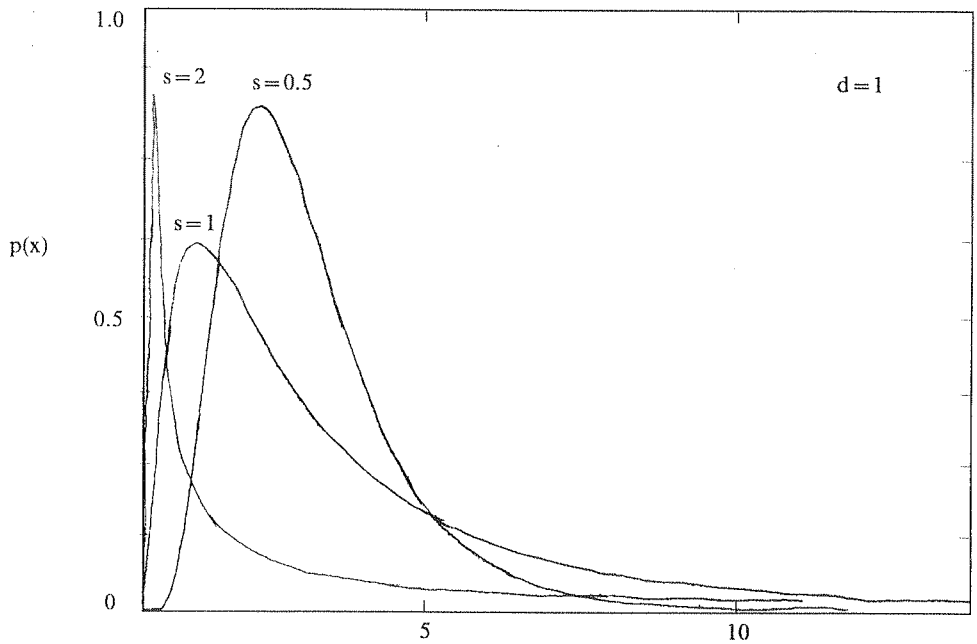
$$p_{LN}(X) = \frac{1}{\sqrt{2\pi}sX} \exp\left[-\frac{\ln^2\left(\frac{X}{d}\right)}{2s^2} \right], \quad X \geqslant 0 \tag{9}$$

gdzie s — parametr kształtu, równy odchyleniu standardowemu zmiennej $ln(X/d)$, d — parametr skali równy medianie zmiennej X.

Wartość średnia m_x i średniokwadratowa $E\{X^2\}$ wynoszą odpowiednio:

$$m_x = d\exp(s^2/2) \tag{10}$$

$$E\{X^2\} = d^2\exp(2s^2) \tag{11}$$



Rys. 2. Rozkłady prawdopodobieństwa amplitudy zakłóceń o modelu logarytmo-normalnym

Na rys. 2 przedstawiono wykresy funkcji gęstości prawdopodobieństwa MLN obliczone dla kilku wartości parametru kształtu s przy tej wartości parametru skali d. Model logarytmo-normalny szczególnie dobrze odzwierciedla właściwości probabilistyczne zakłóceń od powierzchni morza, estymowany z próbek zakłóceń parametr kształtu przyjmuje wartości między 1.3 a 2.0 [9].

Funkcja gęstości prawdopodobieństwa w **modelu Weibulla** (MWB) jest opisana zależnością:

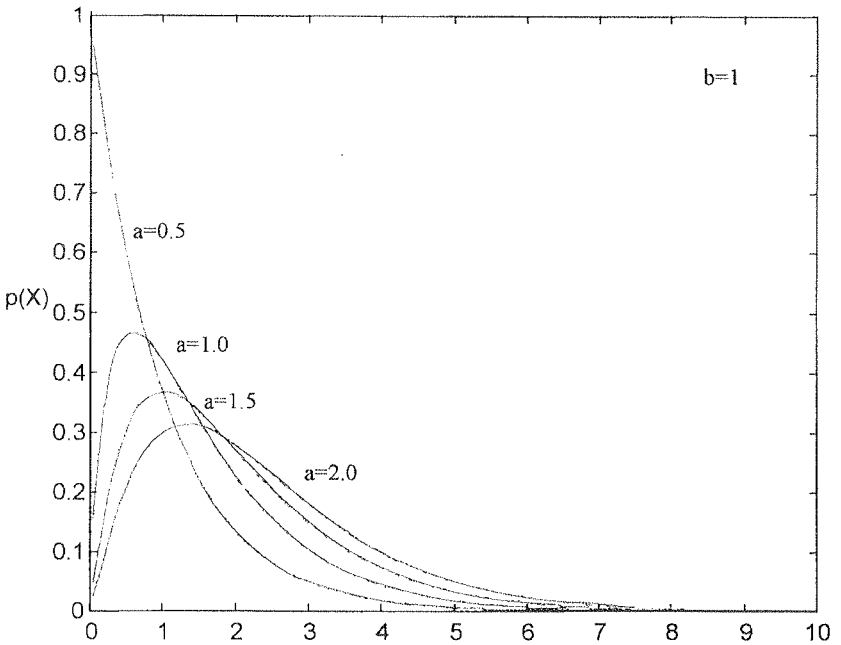
$$p_W(X=\frac{a}{b}\left(\frac{X}{b}\right)^{a-1}\exp\left[-\left(\frac{X}{b}\right)^{a}\right],\ X\geqslant 0 \tag{12}$$

gdzie a — parametr kształtu, b — parametr skali.
Wartość średnia m_x i średniokwadratowa $E\{X^2\}$ wynoszą odpowiedni:

$$m_X=b\Gamma\left(\frac{1}{a}+1\right) \tag{13}$$

$$E\{X^2\}=b^2\Gamma\left(\frac{2}{a}+1\right) \tag{14}$$

gdzie $\Gamma()$ — funkcja gamma Eulera.
Rozkład prawdopodobieństwa dla wybranych wartości parametru kształtu a, przy tym samym parametrze skali b, przedstawia rys. 2.



Rys. 3. Rozkłady prawdopodobieństwa amplitudy zakłóceń o modelu Weibulla

Model Weibulla stosuje się najczęściej do opisu zakłóceń od powierzchni ziemi (parametr kształtu od 0.3 do 0.6) [1] oraz zakłóceń atmosferycznych (parametr kształtu od 1.2 do 2.0) [3]. Warto podkreślić, że dla parametru kształtu $a=2$ rozkład Weibulla staje się rozkładem Rayleigha.

Funkcję gęstości prawdopodobieństwa zakłóceń o **modelu typu K** (MTK) przedstawia zależność:

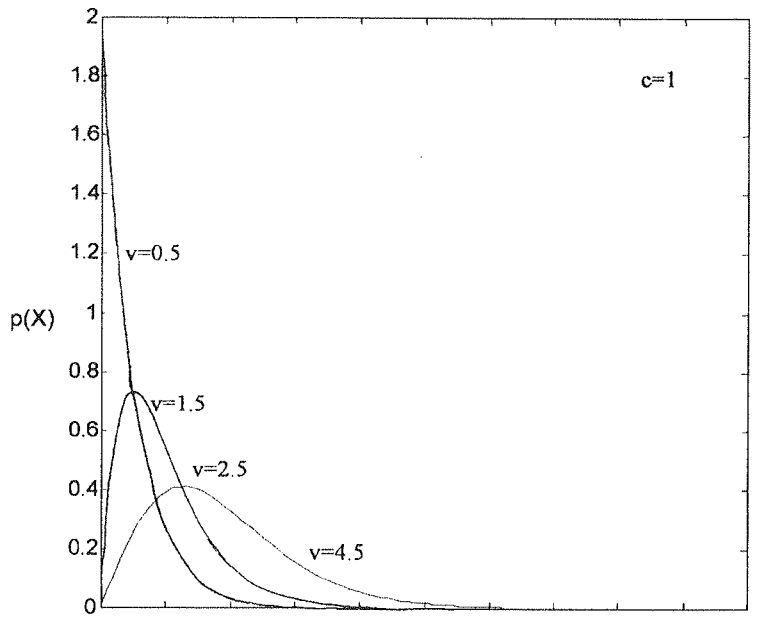
$$p_K(X)=\frac{2c}{\Gamma(v)}\left(\frac{cX}{2}\right)^{v}K_{v-1}(cX),\ X\geqslant 0 \tag{15}$$

gdzie c — parametr skali (c-$2w\sqrt{(\pi/4)}$, v — parametr kształtu, $K_{v-1}()$ — modyfikowana funkcja Bessela (funkcja Mc Donalda) rzędu $v-1$, $\Gamma()$ — funkcja gamma Eulera. Wartość średnia m_x i średniokwadratowa $E\{X^2\}$ wynoszą odpowiednio:

$$m_X=\frac{2\Gamma\left(v+\frac{1}{2}\right)\Gamma\left(\frac{3}{2}\right)}{c\Gamma(v)}, \tag{16}$$

$$E\{X^2\}=\frac{4\Gamma(v+1)\Gamma(2)}{c^2\Gamma(v)} \tag{17}$$

Rys. 4 przedstawia rozkłady prawdopodobieństwa amplitudy zakłóceń MTK dla kilku wartości parametru kształtu, przy tej samej wartości parametru skali.



Rys. 4. Rozkłady prawdopodobieństwa amplitudy zakłóceń o modelu typu K

Model typu K jest najbardziej uniwersalnym z modeli niegaussowskich, dlatego stosuje się go z powodzeniem do opisu zakłóceń od powierzchni morza, powierchni lądu czy też opadów atmosferycznych [1, 11]. W związku z tym parametr kształtu może przyjmować wartości we względnie szerokim przedziale: od 0.1 do 5. Pewną wadą tego modelu jest skomplikowana postać funkcji rozkładu prawdopodobieństwa (15), co utrudnia prowadzenie obliczeń analitycznych.

Przy założeniu stacjonarności procesu, modelującego zakłócenia bierne, a także niezależności statystycznej między próbkami tych zakłóceń, można w stosunkowo prosty sposób generować ciągi losowe o zadanych rozkładach prawdopodobieństwa, symulujące wybrany model. O ile założenie stacjonarności jest słuszne przy krótkich czasach pomiarów i niezbyt dużych obszarach, z których rejestruje się zakłócenia, to przyjęcie niezależności statystycznej jest na ogół zbyt dalego idącym uproszczeniem. Radiolokacyjne zakłócenia bierne charakteryzują się bowiem korelacją czasowo-przestrzenną, określaną między innymi poprzez funkcję autokorelacji i jej transformatą Fouriera, reprezentującą widmo mocy. Współcześnie wykorzystuje się dwa modele widma mocy: Barlowa i Fishbeina [12].

Model Barlowa, stosowany w opisie zakłóceń powstałych w wyniku odbicia od powierzchni ziemi i morza, ma postać:

$$S(f) = S(0)\exp\left(-\frac{f^2}{2\sigma_f^2}\right) \tag{18}$$

gdzie $S(0)$ — wartość widma dla częstotliwości równej 0, σ_f^2 — wariancja widma mocy.

Model Fishbeina, stosunkowo dobrzez opisujący widmo zakłóceń charakterystycznych dla obszaru wielkomiejskiego z gęstą i wysoką zabudową, stanowi zależność:

$$S(f)=\frac{S(0)}{1+|f/f_c|^3} \tag{19}$$

gdzie $S(0)$ — wartość widma dla częstotliwości równej $0, f_c$ — 3- dB szerokość widma.

Stopień skolerowania próbek zakłóceń zależy od wielu czynników, na przykład od rozdzielczości radaru, częstotliwości i polaryzacji fali nośnej, zróżnicowania oświetanej powierzchni itd. Korelacja silnie maleje przy wzroście odległości czasowej między próbkami (już od około 10 ms), a w radarach o przemiennej częstotliwości uzyskuje się istotną dekorelację zakłóceń od impulsu do impulsu sondującego [11]. Takie techniki nie są zawsze dostępne i dlatego pełny proces modelowania zakłóceń musi obejmować także generowanie ciągów próbek o określonej funkcji autokorelacji.

Przy takim podejściu symulacja radiolokacyjnych zakłóceń biernych za pomocą przedstawionych modeli sprowadza się do generowania ciągów losowych $\{x_i, i=1,\ldots,M\}$, będących realizacją dyskretnych procesów stochastycznych o zadanych cechach probabilistycznych, takich jak brzegowe rozkłady prawdopodobieństwa $p(x)$ czy też macierze korelacji $\boldsymbol{R}$. Brzegowy rozkład prawdopodobieństwa wyznacza histogram wartości próbek losowych — każda próbka „otrzymuje" szczególną wartość ze względną częstotliwością, wynikającą z rozkładu. Wzajemne relacje między generowanymi próbkami wynikają z kolei z przyjętej macierzy korelacji $\boldsymbol{R}$, której elementy zdefiniowane jako;

$$R_{i,j}=\frac{E[x_ix_j]-E[x_i]E[x_j]}{\sqrt{D^2(x_i)D^2(x_j)}},\ i,j=1,2,\ldots,M \tag{20}$$

gdzie $E[\,]$ oznacza uśrednienie probabilistyczne a $D^2(z)$ wariancję zmiennej x,
są dyskretnymi wartościami przyjętej funkcji autokorelacji. Tak więc brzegowy rozkład prawdopodobieństwa oraz macierz korelacji symulowanych próbek zakłóceń opisują dyskretną realizację stacjonarnego procesu stochastycznego, modelującego wybrany rodzaj zakłóceń biernych.

3. GENERACJA NIEZALEŻNYCH PRÓBEK ZAKŁÓCEŃ

Najbardziej rozpowszechnioną metodą konstruowania ciągu niezależnych liczb losowych jest tak zwany liniowy algorytm kongruentny [13], za pomocą którego w większości komputerów generuje się sekwencje próbek $\{u_1, u_2, \ldots, u_k\}$ zmiennych losowych U, każda o rozkładzie jednostajnym w przedziale [0, 1], czyli o brzegowej funkcji gęstości prawdopodobieństwa:

$$p(U)=\begin{cases}1 & \text{dla } 0\leqslant U\leqslant 1\\ 0 & \text{dla } U<0 \text{ i } U>1\end{cases}. \tag{21}$$

Ciąg $\{u_i\}$ liczb losowych o powyższym rozkładzie, oznaczanym dalej przez $U(0, 1)$, stanowi bazę do tworzenia ciągu $\{x_i, i=1,...,M\}$ niezależnych próbek zmiennej X o innym, dowolnym rozkładzie prawdopodobieństwa. Algorytmy takich generatorów są opisane w [13, 14] i sprowadzają się do odpowiedniego przetworzenia próbek u_i:

$$x_i = f(u_1, u_2, \ldots, u_k). \tag{22}$$

Postać funkcji f oraz liczba k zależą od wybranej metody generacji — jedną z najbardziej uniwersalnych jest metoda odwracania dystrybuanty [13], która wymaga znajomości dystrybuanty $P(x)$ zmiennej losowej X. Jedna próbka u_i wystarcza do wygenerowania próbki x_i, zgodnie z zależnością:

$$x_i = P^{-1}(u_i) \tag{23}$$

gdzie $P^{-1}()$ jest funkcją odwrotną do $P()$.

Niestety, nie zawsze można znaleźć postać analityczną funkcji P^{-1}. W takich przypadkach zamiast przetwarzania próbek u_i stosuje się na przykład odpowiednie przetwarzanie próbek $\{n_1, n_2,..., n_k\}$ zmiennej losowej N o gaussowskim (normalnym) rozkładzie prawdopodobieństwa, zerowej wartości średniej i wariancji równej 1. Rozkład ten będzie oznaczany dalej jako $N(0, 1)$. Jedną próbkę n_i otrzymuje się z dwóch wygenerowanych próbek u_1 i u_2 poprzez następujące przekształcenie [14]:

$$n_i = \sqrt{-2\ln u_1}\cos(2\pi u_2) \tag{24}$$

Na podstawie powyższych rozważań, biorąc odpowiednie zależności na funkcje transformujące próbki u_i oraz n_i, uzyskuje się następujące algorytmy generacji ciągu niezależnych próbek zakłóceń $\{x_i;\ i=1,...,M\}$ dla zakłóceń MLN i MWB [13, 14]:

Algorytm 1 — Model logarytmo-normalny

W poniższym algorytmie można wyróżnić następujące kroki:

1. Przyjąć założone wartości parametrów d oraz s.
2. Wygenerować ciąg $\{n_i;\ i=1,...,M\}$ niezależnych próbek losowych o rozkładzie $N(0,1)$, korzystając z dostępnego generatora liczb o rozkładzie $U(0,1)$ oraz przekształcenia (24).
3. Przetworzyć każdy element ciągu $\{n_i;\ i=1,...,M\}$ w element ciągu $\{x_i, i=1,...,M\}$ według zależności

$$x_i = d\exp(s\,n_i).$$

Algorytm 2 — Model Weibulla

W poniższym algorytmie można wyróżnić następujące kroki:

1. Przyjąć założone wartości parametrów a oraz b.
2. Wygenerować ciąg $\{u_i;\ i=1,...,M\}$ niezależnych próbek o rozkładzie $U(0,1)$, korzystając z dostępnego generatora.

3. Przetworzyć każdy element ciągu $\{u_i;\ i=1,\ldots,M\}$ w element ciągu $\{x_i,\ i=1,\ldots,M\}$ według zależności:

$$x_i = b(-\ln u_i)^{\frac{1}{a}},$$

W przypadku MTK rozkład prawdopodobieństwa ma na tyle skomplikowaną postać, że każda z przedstawionych metod jest trudna do zastosowania. Dlatego często dla MTK używa się metod iteracyjnych lub ARMA, które są czasochłonne i mało efektywne [5]. Stosunkowo prosty w realizacji jest sposób oparty na pojęciu przestrzennie niezmiennego procesu stochastycznego, który został przedstawiony w następnym rozdziale. Ograniczając odpowiednio algorytm dla skolerowanego MTK otrzymujemy schemat generacji niezależnych próbek tego modelu [18]:

Algortym 3 — Model typu K

W poniższym algorytmie można wyróżnić następujące kroki:

1. Przyjąć założone wartości parametrów c i v.
2. Wygenerować ciąg $\{n_i;\ i=1,\ldots,M\}$ próbek niezależnych o rozkładzie $N(0,1)$, korzystając z dostępnego generatora liczb o rozkładzie $U(0,1)$ oraz przekształcenia (24).
3. Wygenerować próbkę y zmiennej losowej Y o rozkładzie:

$$p(Y) = \frac{2c}{\Gamma(v)2^v}(cY)^{2v-1}\exp\left[-\frac{(cY)^2}{2}\right],$$

poprzez wyznaczenie numeryczne dystrybutanty dla zadanych parametrów c i v oraz zastosowanie metody odwracania dystrybuanty.

4. Wyznaczyć wartość próbki s z zależności:

$$s = \frac{cy}{\sqrt{2v}}.$$

5. Na podstawie ciągu $\{n_i\}$ wyznaczyć ciąg losowy $\{z_i,\ i=1,\ldots,M\}$ według wzoru:

$$z_i = sn_i.$$

6. Wyznaczyć próbki ciągu $\{x_i;\ i=1,\ldots,M\}$ o rozkładzie typu K zgodnie ze wzorem:

$$x_i = z_i + \frac{2\Gamma(v+0{,}5)\Gamma(1{,}5)}{c\Gamma(v)}.$$

4. ALGORYTMY GENERACJI PRÓBEK SKOLEROWANYCH

Modelowanie ciągu skolerowanych próbek zakłóceń $\{x_i,\ i=1,\ldots,M\}$ jest zagadnieniem o wiele bardziej skomplikowanym niż próbek niezależnych. Teoretycznie można byłoby zastosować metodę odwracania dystrybuanty, ale pod warunkiem znajmości łącznego M-wymiarowego rozkładu prawdopodobieństwa tych próbek

[14]. Znalezienie analitycznej postaci tego rozkładu, z wyjątkiem gaussowskiego modelu zakłóceń, jest zadaniem niezwykle trudnym, praktycznie niewykonalnym. Dlatego do wygenerowania sekwencji próbek o zadanych właściwościach korelacyjnych stosuje się dyskretną postać równania Karhunena-Loeve'a, które wiąże ciąg losowy $\{x_i\}$ o określonej macierzy korekcyjnej $\boldsymbol{R}$ z ciągiem losowym $\{z_j, j=1,\ldots,M\}$ reprezentującym niezależne zmienne losowe o zerowej wartości średniej i wariancji równej jeden [14]. Równanie to ma postać:

$$x_i = \sum_{j=1}^{M} h_{i,j} z_j \tag{25}$$

gdzie $h_{i,j}$ są elementami macierzy $\boldsymbol{H}$, związanej z macierzą $\boldsymbol{R}$ zależnością:

$$\boldsymbol{H}\boldsymbol{H}^T = \boldsymbol{R} \tag{26}$$

O ile właściwości korelacyjne ciągu $\{x_i\}$ są jednoznacznie określone poprzez macierz $\boldsymbol{R}$, to zarówno łączny rozkład prawdopodobieństwa, jak i pojedyncze rozkłady brzegowe zależą od postaci rozkładów elementów $\{z_j\}$. Ponieważ każdy element ciągu $\{x_i\}$ jest utworzony zgodnie z zależnością (25), to jego rozkład prawdopodobieństwa jest rozkładem ważonej sumy niezależnych zmiennych losowych. Jeżeli każda z tych zmiennych ma rozkład normalny, to próbka x_i będzie miała także normalny i problem modelowania skorelowanych zakłóceń gaussowskich sprowadza się do znalezienia miacierzy $\boldsymbol{H}$ na podstawie przyjętej macierzy korelacji$\boldsymbol{R}$. Istnieje szereg metod przedstawienia macierzy za pomocą iloczymu dwóch macierzy, będących wzajemnymi transpozycjami. A ponieważ macierz korelacji $\boldsymbol{R}$ jest z definicji symatryczna i dodatnio określona, to przydatnym i skutecznym algorytmem wyznaczania macierzy $\boldsymbol{H}$ jest algorytm, Cholesky'ego [15].

W przypadku modelowania zakłóceń niegaussowskich uzyskanie na podstawie równania (25) próbek o zadanym z góry rozkładzie prawdopodobieństwa jest zadaniem skomplikowanym i nie zawsze możliwym do zrealizowania wprost. Istnieje na przykład metoda, polegająca na stopniowej modyfikacji rozkładów niezależnych próbek z_j, aż do uzyskania wymaganego rozkładu próbki x_i, ale ze względu na liczbę i stopień trudności obliczeń jest ona całkowicie nieprzydatna dla zakłóceń niegaussowskich [14]. Sytuacja jest znacznie prostsza w przypadku modelowania zkałóceń, których próbki można wygenerować przetwarzając odpowiednią sekwencję próbek o rozkładzie normalnym. Dodatkowo należy jeszcze określić związek między elementami $R_{i,j}$ założonej maicerzy korelacji $\boldsymbol{R}$ symulowanego modelu a elemantami $r_{i,j}$ macierzy korelacji sekwencji o rozkładzie normalnym. Dla zakłóceń MLN i MWB zależności te można znaleźć w pracach [16, 17]. Ogólny algorytm symulacji dla tych dwóch modeli jest następujący:

- Wygenerować ciąg $\{n_i, i=1,\ldots,M\}$ próbek niezależnych o rozkładzie $N(0,1)$.
- Na podstawie przyjętej macierzy korelacji $\boldsymbol{R}$ wyznaczyć elementy $r_{i,j}$.
- Stosując algorytm Cholesky'ego wyznaczyć na podstawie macierzy złożonej z elementów $r_{i,j}$ macierz $\boldsymbol{H}$.
- Stosując równanie Karhunena-Loeve'a wyznaczyć elementy skorelowanego ciągu $\{z_i\}$ o rozkładzie normalnym.

- Przetworzyć ciąg $\{z_i\}$ na podstawie odpowiedniej zależności w modelowany ciąg $\{x_i\}$.

Niestety powyższy algorytm nie może być zastosowany dla symulacji zakłóceń MTK, ponieważ nie są znane zależności funkcyjne zarówno przetwarzające próbki o rozkładzie normalnym w próbki o rozkładzie typu K, a także między elementami $R_{i,j}$ oraz $r_{i,j}$. Praktyczna metoda generacji skorelowanych liczb losowych o rozkładzie typu K opiera się na teorii przestrzennie niezmiennych procesów stochastycznych [5, 18]. Jedno z twierdzeń tej teorii mówi, że ciąg losowy $\{x_i\}$ o określonej fukcji gęstości prawdopodobieństwa każdej z próbek oraz o określonej macierzy korelacji $\boldsymbol{R}$ można przedstawić w postaci ciągu skorelowanych próbek gaussowskich $\{z_i\}$ o zerowej wartości średniej i tej samej macierzy korelacji $\boldsymbol{R}$, przy czym każda z tych próbek musi być pomnożona przez próbkę losową s o tak zwanym charakterystycznym (dla danego procesu) rozkładzie prawdopodobieństwa $p_s(s)$. W przypadku próbek zakłóceń MTK istnieje stosunkowo prosta postać rozkładu charakterystycznego [18], co pozwala na opracowanie skutecznego algorytmu symulacji tych zakłóceń.

Przedstawiona powyżej analiza praz przytoczone zależności i metody pozwalają na sformułowanie algorytmów generacji ciągu skorelowanych próbek $\{x_i; i=1,\ldots,M\}$ dla niegaussowskich radiolokacyjnych zakłóceń biernych. Algorytmy dla poszczególnych modeli są następujące [16, 17, 18]:

Algorytm 4 — Model logarytmo-normalny

W poniższym algorytmie można wyróżnić następujące kroki:

1. Przyjąć założone wartości parametrów razkładu d oraz s.
2. Określić elementy $R_{i,j}$ macierzy korelacyjnej $\boldsymbol{R}$ o wymiarach $M \times M$ zgodnie z założoną funkcją autokorelacji, przy czym macierz musi być dodatnio określona, a elementy $R_{i,j}=R_{j,i}$.
3. Wyznaczyć elementy $r_{i,j}$ z zależności:

$$r_{i,j}=\frac{1}{s^2}\ln\{R_{i,j}[\exp(s^2)-1]+1\}.$$

4. Wyznaczyć elementy macierzy $\boldsymbol{H}$ z poniższych zależności:

$$h_{i,1}=r_{i,1}\sqrt{r_{1,1}} \qquad \text{dla } 1\leqslant i\leqslant M$$

$$h_{i,i}=\sqrt{r_{i,i}-\sum_{k=1}^{i-1}h_{i,k}^2} \qquad \text{dla } 1<i\leqslant M$$

$$h_{i,j}=[r_{i,j}-\sum_{k=1}^{j-1}h_{i,k}h_{j,k}]/h_{j,j} \qquad \text{dla } 1<j<i\leqslant M$$

$$h_{i,j}=0 \qquad \text{dla } i<j\leqslant M.$$

5. Wygenerować ciąg $\{n_j,\ j=1,\ldots,M\}$ próbek niezależnych o rozkładzie $N(0,1)$ korzystając z dostępnego generatora liczb o rozkładzie $U(0,1)$ oraz przekształcenia (24).

6. Wyznaczyć próbki skorelowanego ciągu normalnego $\{z_i;\ i=1,\ldots,M\}$ z zależności:

$$z_i=\sum_{j=1}^{M} h_{i,j}n_j .$$

7. Wyznaczyć próbki skorelowanego ciągu $\{x_i;\ i=1,\ldots,M\}$ o rozkładzie logarytmo-normalnym zgodnie ze wzorem:

$$x_i=d\exp(s\,n_i) .$$

Algorytm 5 — Model Weibulla

W poniższym algorytmie można wyróżnić następujące kroki:

1. Przyjąć założone wartości parametrów d i s.
2. Określić elementy $R_{i,j}$ macierzy korelacyjnej $\boldsymbol{R}$ o wymiarach $M\times M$ zgodnie z założoną fukcją autokorelacji, przy czym macierz musi być dodatnio określona, a elementy $R_{i,j}=R_{j,i}$.
3. Wyznaczyć elementy $r_{i,j}$ z zależności:

$$R_{i,j}=\frac{1}{b^2}\Gamma^2(1+a^{-1})[{}_2F_1(-a^{-1},-a^{-1},1:r_{i,j}^2)-1]$$

4. Wyznaczyć elementy macierzy $\boldsymbol{G}$ z poniższych zależności:

$$h_{i,1}=r_{i,1}\sqrt{r_{1,1}} \qquad \text{dla } 1\leqslant i\leqslant M$$

$$h_{i,i}=\sqrt{r_{i,i}-\sum_{k=1}^{i-1} h_{i,k}^2} \qquad \text{dla } 1<i\leqslant M$$

$$h_{i,j}=[r_{i,j}-\sum_{k=1}^{j-1} h_{i,k}h_{j,k}]/h_{j,j} \qquad \text{dla } 1<j<i\leqslant M$$

$$h_{i,j}=0 \qquad \text{dla } i<j\leqslant M .$$

5. Wygenerować dwa niezależne ciągi $\{n_{1i};\ i=1,\ldots,M\}$ oraz $\{n_{2i};\ i=1,\ldots,M\}$ niezależnych próbek losowych o rozkładzie $N(0,1)$, korzystając z dostępnego generatora liczb o rozkładzie $U(0,1)$ oraz przekształcenia (24).
6. Wyznaczyć próbki dwóch skorelowanych ciągów normalnych $\{z_{1i};\ i=1,\ldots,M\}$ oraz $\{z_{2i};\ i=1,\ldots,M\}$ z zależności:

$$z_{1i}=\sum_{j=1}^{M} h_{i,j}n_{1j} .$$

$$z_{2i}=\sum_{j=1}^{M} h_{i,j}n_{2j} .$$

7. Wyznaczyć próbki skorelowanego ciągu $\{x_i;\ i=1,\ldots,M\}$ o rozkładzie Weibulla zgodnie ze wzorem:

$$x_i = b\left(\frac{z_{1i}^2+z_{2i}^2}{2}\right)^{\frac{1}{a}}$$

Algortym 6 — Model typu K

W poniższym algorytmie można wyróżnić następujące kroki:

1. Przyjąć założone wartości parametrów c i v.
2. Wygenerować ciąg $\{n_i;\ i=1,\ldots,M\}$ próbek niezależnych o rozkładzie $N(0,1)$, korzystając z dostępnego generatora liczb o rozkładzie $U(0,1)$ oraz przekształcenia (24).
3. Wygenerować próbkę y zmiennej losowej Y o rozkładzie:

$$p(Y)=\frac{2c}{\Gamma(v)2^v}(cY)^{2v-1}\exp\left[-\frac{(cY)^2}{2}\right],$$

poprzez wyznaczenie numeryczne dystrybuanty dla zadanych parametrów c i v oraz zastosowanie metody odwracania dystrybuanty.

4. Wyznaczyć wartość próbki s z zależności:

$$s=\frac{cy}{\sqrt{2v}}.$$

5. Na podstawie ciągu $\{n_i\}$ wyznaczyć ciąg losowy $\{z_i,\ i=1,\ldots,M\}$ według wzoru:

$$z_i = sn_i.$$

6. Określić elementy $R_{i,j}$ macierzy korelacyjnej $\boldsymbol{R}$ o wymiarach $M\times M$ zgodnie z założoną funckją autokorelacji, przy czym macierz musi być dodatnio określona, a elementy $R_{i,j}=R_{j,i}$.
7. Wyznaczyć elementy macierzy $\boldsymbol{H}$ z poniższych zależności:

$$h_{i,1}=R_{i,1}\sqrt{R_{1,1}} \qquad \text{dla } 1\leqslant i\leqslant M$$

$$h_{i,i}=\sqrt{R_{i,i}-\sum_{k=1}^{i-1}h_{i,k}^2} \qquad \text{dla } 1<i\leqslant M$$

$$h_{i,j}=[R_{i,j}-\sum_{k=1}^{j-1}h_{i,k}h_{j,k}]/h_{j,j} \qquad \text{dla } 1<j<i\leqslant M$$

$$h_{i,j}=0 \qquad \text{dla } i<j\leqslant M.$$

8. Wyznaczyć próbki skorelowanego ciągu $\{x_i;\ i=1,\ldots,M\}$ o rozkładzie typu K zgodnie ze wzorem:

$$x_i=\sum_{j=1}^{M}h_{i,j}z_j+\frac{2\Gamma(v+0{,}5)\Gamma(1{,}5)}{c\Gamma(v)}.$$

5. PODSUMOWANIE

W artykule opisano trzy najbardziej użyteczne i szeroko stosowane modele matematyczne radiolokacyjnych zakłóceń biernych. Spośród wielu istniejących algorytmów symulacji tych zakłóceń wybrano i zaprezentowano takie, które umożliwiają uzyskiwanie zakładanych funkcji gęstości prawdopodobieństwa generowanych próbek oraz macierzy korelacji, charakteryzującej zależności statystyczne między próbkami. Pozwala to na realizację dyskretnego w czasie procesu stochastycznego, reprezentującego założony model zakłóceń. Dzięki temu przedstawione algorytmy symulacji mogą być z powodzeniem zastosowane przy projekotowaniu odbiorników radiolokacyjnych.

Mimo przeglądowego charakteru artykułu procedury symulacyje zostały opisane bardziej szczegółowo, niż w materiałach źródłowych, bowiem zamierzeniem autora było takie ich opracowanie, żeby mogły być bezpośrednio zaimplementowane w formie procedur komputerowych.

Warto podkreślić, że zastosowanie przedstawionych algorytmów nie ogranicza się tylko do generowania jednowymiarowych ciągów próbek zakłóceń. Projektując odpowiednio macierz korelacji można przetransformować ciąg jednowymiarowy na obszar dwuwymiarowy, według zasad, przedstawionych w pracy [19], tworząc pole zakłóceń o założonych współczynnikach korelacji poziomej i pionowej.

BIBLIOGRAFIA

1. J. B. Billingsley i in.: *Statistical analyses of measured radar ground clutter data.* IEEE trans. AES, vol. AES-35, No 2, 1999, pp. 579–592
2. K. D. Ward, C. J. Baker, S. Watts: *Radar scattering from the ocean surface.* IEE Proceedings, vol. 137, Pt. F, No 2, April 1990, pp. 51–62.
3. M. Sekine i in.: *Non-Rayleigh weather clutter.* IEE Proceedings, vol. 137, Pt. F, No 6, 1980, pp. 471–474.
4. S. Haykin i in.: *Maximum Entropy Spectral Analysis of Radar Clutter.* Proc. IEEE, vol. 70, Sept. 1982, pp. 953–962.
5. E. Conte i in.: *Modelling and simulation of non-Rayleigh radar clutter.* IEE Proceedings, vol. 138, Pt. F, No 2, April 1991, pp. 121–132
6. J. K. Jao, M. Elbaum: *First-order statistics of a non-Rayleigh fading signal and its detection.* Proceedings of the IEEE, vol. 66, 1978, pp. 781–789.
7. E. J. Barton: *Radars Volume5: Radar Clutter.* Artech House, Dedham, Massachusetts, 1975.
8. F. Fay, J. Clarke, R. Peters: *Weibull distribution applied to sea clutter.* Int. Conf. RADAR'77, London, 1977, IEE Conf. Publ. 155, pp. 101–104.
9. A. Farina i in.: *High resolution sea clutter data: A statistical analysis of recorded live data.* IEE Proceedings, vol. 144, Pt. F, No 3, June 1997, pp. 121–130.
10. E. Jakeman, P. N. Pusey: *A Model for Non-Rayleigh Sea Echo.* IEEE Trans. AP, vol. AP-24, 1976, pp. 806–814.
11. S. Watts: *Radar detection prediction in sea clutter using the compoud K-distributed model.* IEE Proceedings, vol. 132, Pt. F, No 7, 1985, pp. 613–620.
12. M. Skolnik: *Radar Handbook.* New York, McGrwa-Hill, 1990
13. R. Zieliński: *Generatory liczb losowych.* Warszawa, WNT, 1972.

14. G. E. Johnson: *Constructions of Particular Random Processes.* Proceedings of the IEEE, vol. 82, No 2, Feb. 1994, pp. 270 – 285
15. E. M. Scheuer, D. C. Stoller: *On the Generation of Normal Random Vectors.* Technometrics, No 4, 1962, pp. 278 – 281.
16. W. J. Szajnowski: *Generation of Correlated Log-Normal Clutter Samples.* Electronics Letters, Vol. 12, No 19, Sept. 1976, pp. 497 – 498
17. W. J. Szajnowski: *The generation of correlated Weibull clutter for signal detection problems.* IEEE Trans. AES, vol. AES-13, No 5, 1977, pp. 536 – 540
18. M. Rangaswamy, D. Weiner, A. Öztürk: *Computer generation of correlated non-gaussian radar clutter.* IEEE Trans. on AES, Vol. 31, No 1, January 1995, pp. 106 – 115.
19. A. Jakubiak, M. Muczko, A. Zieliński: *Wybrane metody modelowania komputerowego zakłóceń biernych.* Przegląd telekomunikacyjny i Wiadomości Telekomunikacyjne, Nr 4, 2000, ss. 298 – 303

A. JAKUBIAK

COMPUTER SIMULATION ALGORITHMS OF RADAR CLUTTER

Summary

Selected methods of clutter modeling for computer simulation purposes are presented in the paper. Non-Gaussian statistics of clutter amplitude, like Log-normal, Weibull and K-distribution, are described. The method enables to generate particular sets of random samples with desired first marginal distribution and to create correlation properties of modelled signal. The paper is essentially tutorial, emphasizing aspects of theory necessary to engineering modeling, and collecting algorithm constructions.

Key words: radar clutter, modelling, computer simulation

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2000, 46, z. 2, ss. 255–272

The implementation of image processing algorithms for the multiprocessor system

MIROSŁAW GAJER

Akademia Górniczo-Hutnicza im. S. Staszica w Krakowie
Katedra Automatyki

Otrzymano 2000.03.01
Autoryzowano 2000.04.22

In the paper the usage of the TMS320C80 Texas Instruments multiprocessor chip for the parallel image processing is describe. In the real-time implementations of image processing algorithms the performance time is a critical parameter, so very often multiprocessor solutions must be used. The TMS320C80 is composed of one master RISC processor and four parallel DSP processors specialised for effcient image processing. Because these processors are quite loosely coupled and they communicate through the common memory, it is possible for this system to implement many different types of multiprocessor architecture. In the paper are presented the results obtained during the implementation of the chosen image processing algorithms for the different architectures such as SIMD, MIMD, MISD and pipeline structure. The attention is paid to the problem of the matching image processing algorithm to the proper multiprocessor architecture in order to minimise the computation time.

Key words: multiprocessor system, parallel computing, image processing

1. DIGITAL IMAGE PROCESSING

The applications related to the field of digital image processing and recognition require the great computation power. The reason is the great amount of information contained in the digital image. Thus, some elementary image processing operations such as convoluation with the mask of 3×3 filter must be repeted for many thousands of pixels. In case of the real-time image processing (robots vision systems, navigation systems) the usage of processors, which posses enough computation power is essential. This is a reason for more and more often usage of multiprocessor devices in the area of digital image processing.

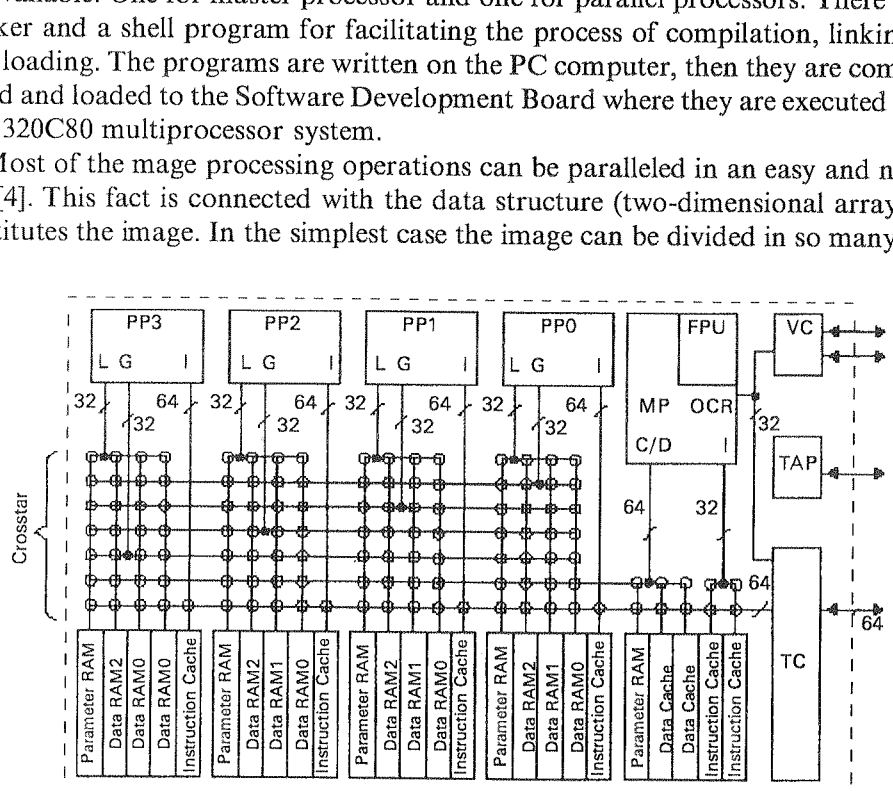
The image processing operations exist in all automatic pattern recognition systems that are often used as robots vision systems. In such systems for the purpose of which is the elements type recognition for montage operations, the image must be

at the beginning preprocessed in order to improve its quality or to extract some features improtant from the point of view of the recognition process. Because robots vision systems most often works as real-time systems fast performing of image pre-processing operations is a very important factor [1], [2], [3].

Texas Instruments was the first that delivered for the market a multiprocessor DSP system in a single chip. The multiprocessor device the TMS320C80 contains four fixed-point, 32-bit DSP processors with the architecture specialised for the very fast performing of image processing operations.

The architecture of the TMS320C80 was shown in Fig. 1. There is a master, 32-bit RISC processor MP and four parallel DSP processors PP0, PP1, PP2, PP3. There is also the video controller VC, which manages image acquistion process and the transfer controller for the communication with the outside world. The TMS320C80 was integrated with the data acquisition blocks and memory blocks, as a PCI board — the Software Development Board. As a code generation tools ANSI C compilers are available. One for master processor and one for parallel processors. There is also a linker and a shell program for facilitating the process of compilation, linking and code loading. The programs are written on the PC computer, then they are compiled, linked and loaded to the Software Development Board where they are executed by the TMS320C80 multiprocessor system.

Most of the mage processing operations can be paralleled in an easy and natural way [4]. This fact is connected with the data structure (two-dimensional array) that constitutes the image. In the simplest case the image can be divided in so many parts



Legend:
L Local port
I Instruction port
OCR On-chip register port
TAP Test access port
VC Video controller
PP0-3 Parallel processors 0-3
G Global port
FPU Floating-point unit
C/D Cacha/data port
TC Transter controller
MP Master processor

Fig. 1. TMS320C80 archtecture block diagram (image from the Texas Instruments web page: www.ti.com)

as many processors one has. Each processor performs the same elementary operations for its fragment of the image. In some cases these fragments must overlap by the width of one on more pixels, which is directly connected with the dimensions of the used filters masks. On the other hand some image processing operations are often multistage processes that are compsed of many operations that follow each other. In such case it is possible to implement these operations for some pipeline architectures. It is also possible some combination of these two different methods of image processing operations paralleling. For the real-time systems where the computation time is essential, the choice of proper multiprocessor architecture is a very important factor. In the next part of this paper some examples of implementations of the chosen image processing operations for the different multiprocessor structures, based on the TMS320C80, are presented.

2. SIMD ARCHTECTURE

The operations such as image filtering or edge detection by the usage of 3×3 filter masks can be directly implemented for the SIMD (Single Instruction Multiple Data) architecture based on some number of the DSP processors of the TMS320C80 multiprocessor chip.The analysed image is divided in so many parts as many DSP processors are used. Each processor operates on its fragment of the image independently of the other processors. The image acquired form the camera is placed in the common external memory of the system. During the access to this memory some contentions may occur, but the transfer controller TC automatically prioritises memory access demands. The master processor MP always takes care of the image acquisition process and the parallel processors PP are used for the image processing and analysis. As an example a mediam filtering operation was implemented for SIMD architectures built of the different number of DSP processors. The median filtering is an operation during which pixels are sorted according to their grey level. As the value of the pixel in the processed image a median value of the sorted pixels is chosen. The median filtering is used for moise elimination from the image. The experiments were carried for 5 different median filter masks, composed of 5, 9, 13, 21 and 25 pixels

Computation time

Table 1

	Number of processors			
pixels	1	2	3	4
5	1,41s	0,75s	0,56s	0,45s
9	3,22s	1,75s	1,34s	1,24s
13	11,95s	7,10s	5,72s	5,35s
21	24,05s	13,21s	10,13s	9,85s
25	31,15s	16,97s	12,72s	10,86s

Table 2

Speedup coefficient

	Number of processors			
pixels	1	2	3	4
5	1.00	1.88	2.52	3.13
9	1.00	1.84	2.4	2.59
13	1.00	1.63	2.08	2.23
21	1.00	1.82	2.36	2.44
25	1.00	1.83	2.45	2.87

respectively. In the Tab. 1 the image processing times are given and in the Tab. 2 the computation speedup coefficient values are given, depending on the number of processors used and the number of the pixels of the filter mask.

Obtained for different numbers of DSP processors speedup coefficient values are much lower than their maximal theoretically possible values that is the number of processors used in SIMD architecture. This is caused by the communication overheads occurring during the work of the TMS320C80 system. Higher values of speedup coefficients can be obtained while the program codes for PPs can be whole placed in their cache memories. In such situation there is no cache refreshing necessary and the queue at the transfer controller TC is much shorter. An example image obtained after median filtering was shown in Fig. 2.

Fig. 2. Image after median filtering (obtained by the usage of the TMS320C80 system)

3. MIMD ARCHTECTURE

As the example of the implementation of image processing operation for MIMD (Multiple Instruction Multiple Data) architecture the image histogram equalisation will be presented. The image histogram equalisation is the operation that is used in order to improve the image contrast. In the image with the equalised histogram the full range of grey levels is used. The algorithm of histogram equalisation is a two-stage algoritm [5]. In the first step the image histogram is calculated. During this operation for each grey level number of possessing it pixel is calculated. Then the array of 256 numbers is composed **histogram[.]**. Each element of this array, denoted as **histogram [i]** tells the number of pixels of the grey level **i**. In the next step the lookup table **LUT[.]** is composed.

$$LUT[k] = \frac{\sum_{i=0}^{k} histogram[i]}{n} \tag{1}$$

Each element of the lookup table **LUT[k]** determinates a new grey level value for the pixels that had in the acquired from the camera image the grey level **k**. The total number of pixels contained in the image was denoted by **n**.

The image histogram equalisation was implemented for the MIMD architecture composed of 5 processors of the TMS320C80 muliprocessor system. First the master processor MP acquires the image from the camera and places it in the buffer in system external memory. Then each parallel processor PP0, PP1, PP2 and PP3 calculates partial histogram for the given fragment of the image. The master prcessor sums

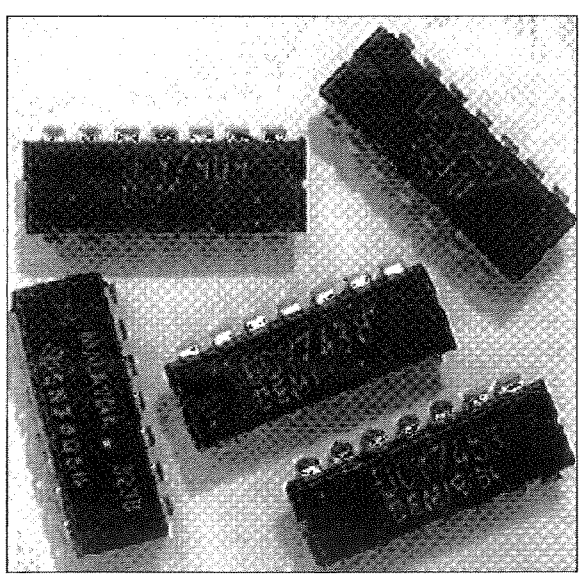


Fig. 3. Image with equalised histogram (obtained by the usage of the TMS320C80 system)

up these partial histograms and so the full image histogram is calculated. Based on this histogram the master processor calculates the lookup table. The lookup table is passed to each of parallel processors. Based on the lookup table each of parallel processors equalises the histogram of its fragment of the image. An example image with the equalised histogram was shown in Fig. 3.

4. MISD ARCHITECTURE

In the MISD (Multiple Instruction Single Data) architecture each processor performs different operations on the same data set. In many multiprocessor classification systems MISD architecture is enumerated with a remark that such type of architecture was defined only for the sake of classification system completeness, because it is difficult in practice to find a computer working on the base of MISD architecture [6], [7], [8], [9]. In case of image processing system the above conclusion heed not be true, because in some image preprocessing systems there is very often necessity of making many different operations on the same image. For such puropose the MISD architecture matches ideally.

As an example of implemantation of image processing operations for the MISD architecture the edge detection operation will be considered. The edges must be detected in four different directions (north, south, west and east) by the medium of four different filter masks. Each of the parallel processors of the TMS320C80 calculates one of the directional edges. Finally one obtains four images with detected edges, each in different direction. An example image with detected edges in north direction was shown in Fig. 4.

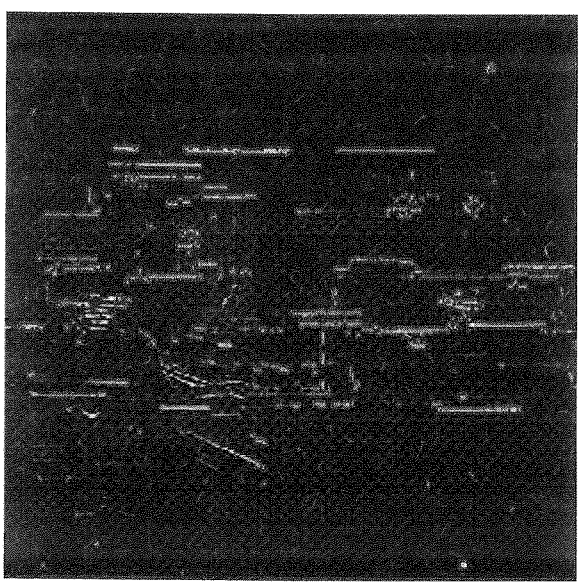


Fig. 4. Image with detected edges in north direction (obtained by the usage of the TMS320C80 system)

5. PIPELINE STRUCTURE

The computer image processing is very often a multistage process composed of many consecutive stages. As an example an algorithm detecting contours of the objects contained in the image. The considered algorithm is composed of four basic image processing operation. The image acquired from the camera in the first step undergoes low-pass filtering during which it is smoothed and noise is also eliminated. The mask of the used low-pass filter was shown in Fig. 5.

1	2	1
2	4	2
1	2	1

Fig. 5. Mask of the low-pass filter

In the next step image is thresholded and each pixel gets black or with depending on that if its grey level is yelow or above threshold value. Then edges from the binary image are detected by the usage of the Laplace filter. The mask of the filter was shown is Fig. 6.

	-1	
-1	4	-1
	-1	

Fig. 6. Mask of the Laplace filter

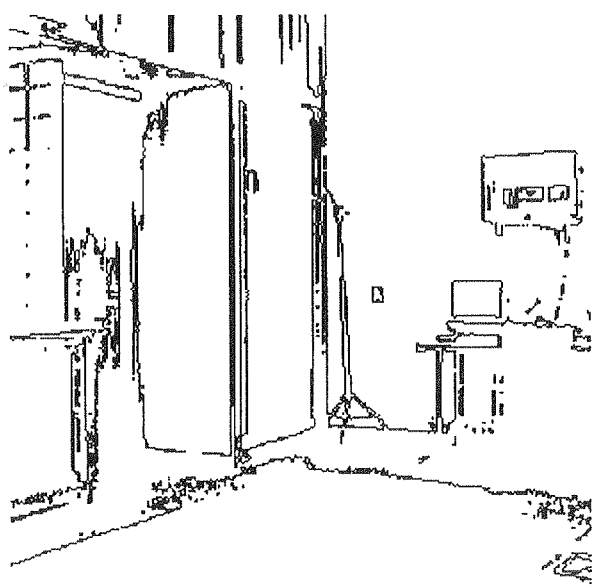


Fig. 7. Image with detected contours (obtained by the usage TMS320C80 system)

In the last step the image with detected contours undergoes logic filtering in the purpose of elimination of single black pixels placed on the white background. The image obtained as a results of performing these operation was shown in Fig. 7.

For the sake of parallel performing of the above operations a five-stage pipeline was used. The first stage of this pipeline constitutes the master processor that takes control of the image acquistion process. Further pipeline stages were implemented for 4 parallel processors, according to the Tab. 3.

Table 3

Pipeline structure

Stage	Processor	Operation
1	MP	Image acquisition
2	PP0	Low-pass filtering
3	PP1	Thresholding
4	PP2	Laplace filtering
5	PP3	Logic filtering

6. HYBRIDE ARCHITECTURE

The TMS320C80 is the Texas Instruments first multiprocessor system in a single chip. Its architecture is optimised to the very fast performing of image processing operations [10]. Due to its special features such as loosely coupled processors with the communication mechanism by the medium of common both external and internal memories, many different multiprocessor configurations can be organised using the different number of its processors. In the case of real-time systems the performance time of image processing operations is a very important factor. Thus for each image processing system a proper multiprocessor configuration must be chosen in order to minimise computation time. Because the TMS320C80 is equipped with five processors pipeline structures up to five stages can be implemented for it. In case when the pipeline stage number is lower than five, some of them can be realised as a SIMD architecture. Such multiprocessor configuration is called hybrid architectire. As a SIMD architecture should be implemented these pipeline stages that require the longest computation time, because the work of the other pipeline stages must be suspended due to the system synchronisation.

An example scheme of some hybrid architecture was shown in Fig. 8. This is the pipeline architecture that is composed of three stages. The first stage of this pipeline is the master processor MP, which takes care of the image acquisition. In the second stage the median filtering is done. This operation is performed simultaneously by two of the parallel processors PP0 and PP1. In the third stage of this pipeline edges are detected by the use of isotropic filtering. For the isotropic filtering thw different filter masks must be used. One filter mask must be used for the horizontal and one for the vertical filtering. These masks were shown in Fig. 9 and Fig. 10.

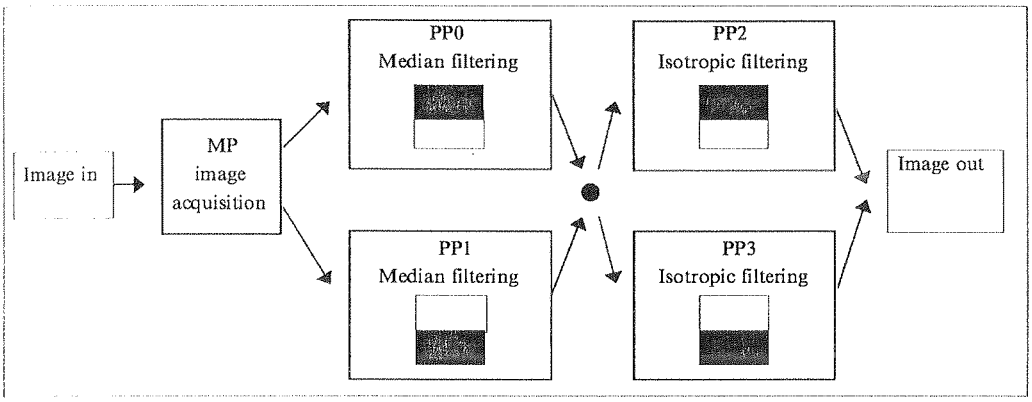


Fig. 8. Hybrid architecture

1	$\sqrt{2}$	1
0	0	0
-1	$-\sqrt{2}$	-1

Fig. 9. Horizontal mask

-1	0	1
$-\sqrt{2}$	0	$\sqrt{2}$
-1	0	1

Fig. 10. Vertical mask

The pixels of the image obtained during these two different filtering processes are raised to power two and summed up and then from the obtained result a square root is calculated. The obtained value is a value of the pixel in the final image with detected edges. An example image, which is the product of the above hybrid architecture was shown in Fig. 11.

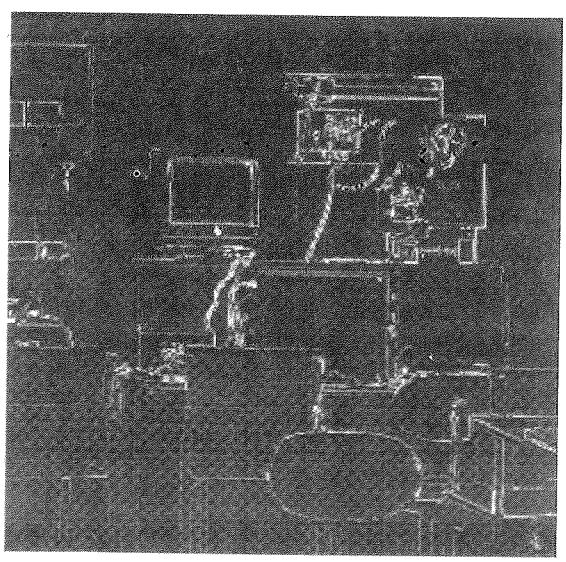


Fig. 11. Image with detected edges (obtained by the usage of the TMS320C80 system)

The other hibride pipeline architecture scheme was shown in Fig. 12. This multiprocessor structure is used for the contours detection of the objects contained in the image. The pipeline is also composed of three stages. First of which is used for image acquisition, second for median filtering and the third Laplace filtering. Because the median filtering is the longest lasting operation it was implemented for the SIMD architecture constructed from three parallel processors: PP0, PP1 and PP2.

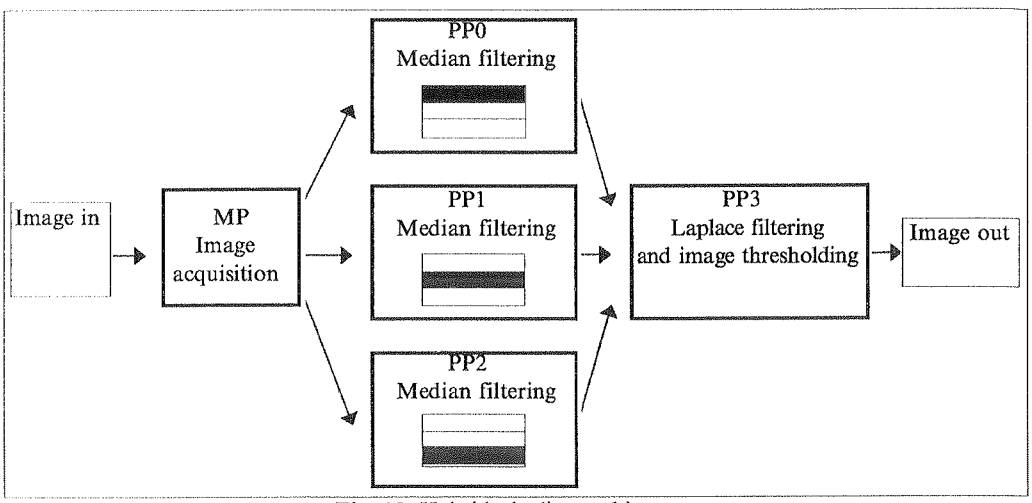


Fig. 12. Hybrid pipeline architecture

-1	-1	-1
-1	8	-1
-1	-1	-1

Fig. 13. Laplace filter mask

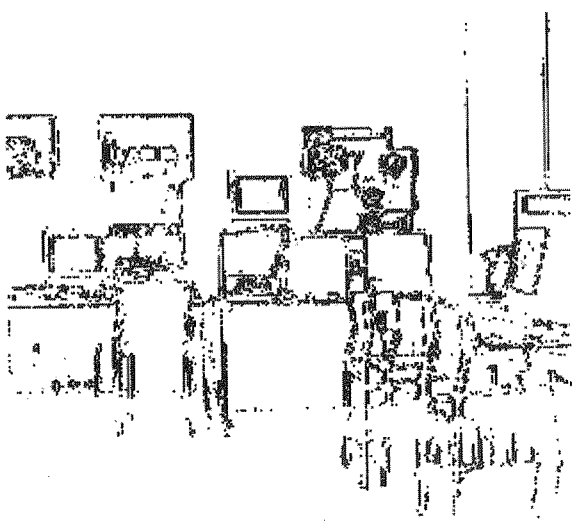


Fig. 14. Image with detected edges and contours (obtained by the usage of the TMS320C80 system)

The image after median filtering undergoes the Laplace filtering by the usage of filter mask shown in Fig. 13. If the pixel value, which is obtained after filtering, is greater than some threshold value it means that the boarder of the object in the image was detected. Such pixel obtains a black colour. In the other case pixel gets white.

An example image obtained as a result of the work of the aboved multiprocessor hybrid architecture was shown in Fig. 14. The edgas and conturs of the objects were detected, while the image background became white.

7. M-SIMD ARCHITECTURE

According to [11] all the multiprocessor architecture can be classified into MIMD, SIMD of MISD architectures. The presented aboved pipeline and hybrid structures are special cases of MIMD architecture, which work in some special way, so in the paper they were distinguished from the other kinds of MIMD architectures. There are also known in the subject literature other types of multiprocessor architectures, which complete the aboved classification [12], [13], [14]. One fo them is the M-SIMD (Multiple — Single Instruction Multiple Data) architecture that was proposed for the first time proposed in [15]. Such architecture is a combination of a few SIMD architectures. Each of thesee architectures performs different kind of operations.

As an example of M-SIMD architecture Robert's edge detection algorithm implementation for the TMS320C80 will be distribed. For this operation a 9-element filter mask must be used. That mask was presented in Fig. 15.

p1	p2	p3
p4	p5	p6
p7	p8	p9

Fig. 15. Robert's edge detection filter mask

The values of the pixels in processed image are calculated following the formula:

$$x=\sqrt{(p5-p9)^2+(p6-p8)^2} \tag{2}$$

The computions can be implemented for M-SIMD architecture composed of two SIMD architectures in such way that the first of them (PP0, PP1) calculated the image

0	0	0
0	1	0
0	0	-1

Fig. 16. Filter mask no. 1

0	0	0
0	0	1
0	-1	0

Fig. 17. Filter mask no. 2

using for this purpose filter mask no. 1, while the second (PP2, PP3) uses for image processing filter mastk no. 2. These filter masks were shown in Fig. 16 and 17 respectively.

In the last step the master processor MP of the TMS320C80 calculates the square root of the sum of second powers of obtained during filtration values. The described M-SIMD architecture cheme was presented if Fig. 18. An example image the Robert's edge detection operation was presented in Fig. 19.

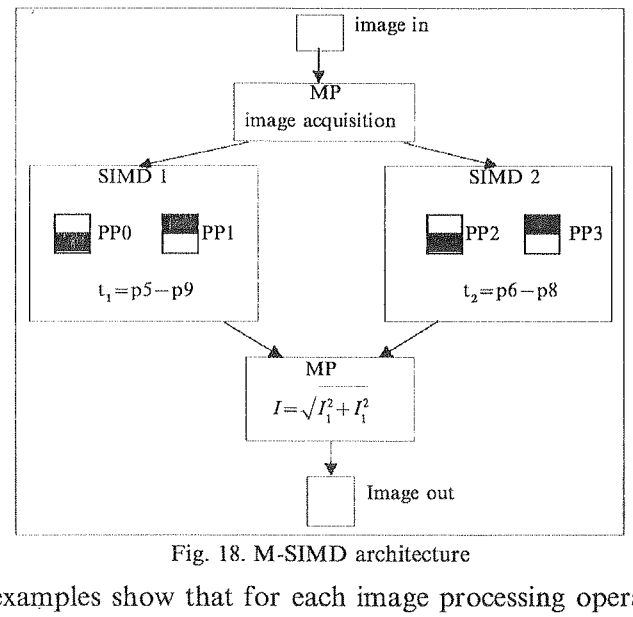


Fig. 18. M-SIMD architecture

The above examples show that for each image processing operation the proper multiprocessor configuration must be found. In some cases it can be a simple SIMD

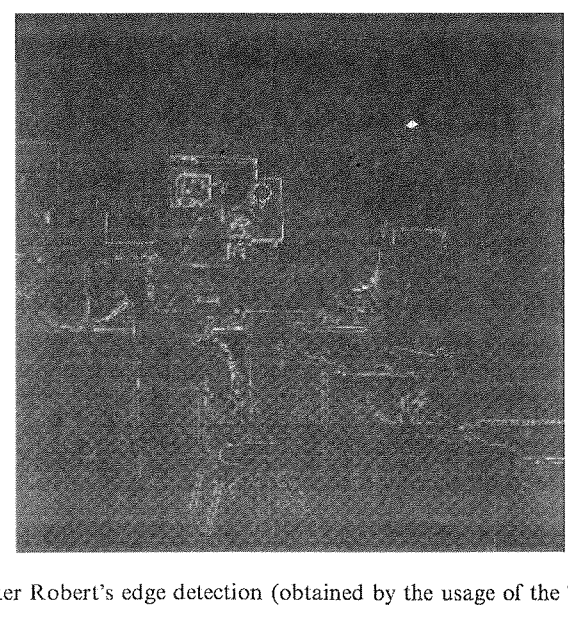


Fig. 19. Image after Robert's edge detection (obtained by the usage of the TMS320C80 system)

architectures, where all processors of the system perform the same operation for different set of pixels. But in some cases some more sophistaticated multiprocessor architecture must be configured. The flexibility of the TMS320C80 structure allows for configuration of different type of MIMD architecture that special cases are pipeline structures proper for multistage image processing operations.

8. IMPLEMENTATION OF FFT ALGORITHM

In some cases image analysis must be performed in the frequency domain. Twodimensional Fourier transform is the tool that allows for image conversion from space to frequency domain. In order to make the computation time shorter fast Fourier transform algorithm FFT must be implemented.

FFT algorithm is based on the butterfly scheme [55] that easily undergoes parallelisation. The butterfly scheme is divided in four parts that are assingned each to the other DSP processor in the way shown in Fig. 20. Thus the FFT computations are divided in two stages. In the first stage each DSP processor performs calculations for its own input data and for data obtained from other processors during inter-processor communication, it also sends some output date to the other processors. In the second stage there is no longer inter-processor communication and each processor is computing only its own portion of data.

Total FFT computations time

Table 4

	Computation time [2]			Speed-up coefficient	
n-point transform	1 processors	2 processors	4 processors	2 processores	4 processors
16	$2{,}26 \cdot 10^{-3}$	$2{,}08 \cdot 10^{-3}$	$1{,}78 \cdot 10^{-3}$	1,08	1,27
32	$5{,}62 \cdot 10^{-3}$	$4{,}91 \cdot 10^{-3}$	$4{,}13 \cdot 10^{-3}$	1,14	1,36
64	$1{,}36 \cdot 10^{-2}$	$1{,}14 \cdot 10^{-2}$	$9{,}01 \cdot 10^{-3}$	1,19	1,51
128	$3{,}18 \cdot 10^{-2}$	$2{,}36 \cdot 10^{-2}$	$1{,}93 \cdot 10^{-2}$	1,34	1,64
256	$7{,}67 \cdot 10^{-2}$	$5{,}15 \cdot 10^{-2}$	$4{,}15 \cdot 10^{-2}$	1,48	1,85

The total computation time for FFT implementation for one, two and four DSP processors was given in Tab. 4.

Looking at Tab. 4. one can see that speed-up coefficients do not achieve big values. This fact is caused by relatively big inter-processor communication overheads that take place during the first stage of butterfly algorithm. One can also see that the value of speed-up coefficient is growing with the number of transform points. This connected with the fact that for bigger portions of input data the ratio of computation time to the inter-processor communication time is greater.

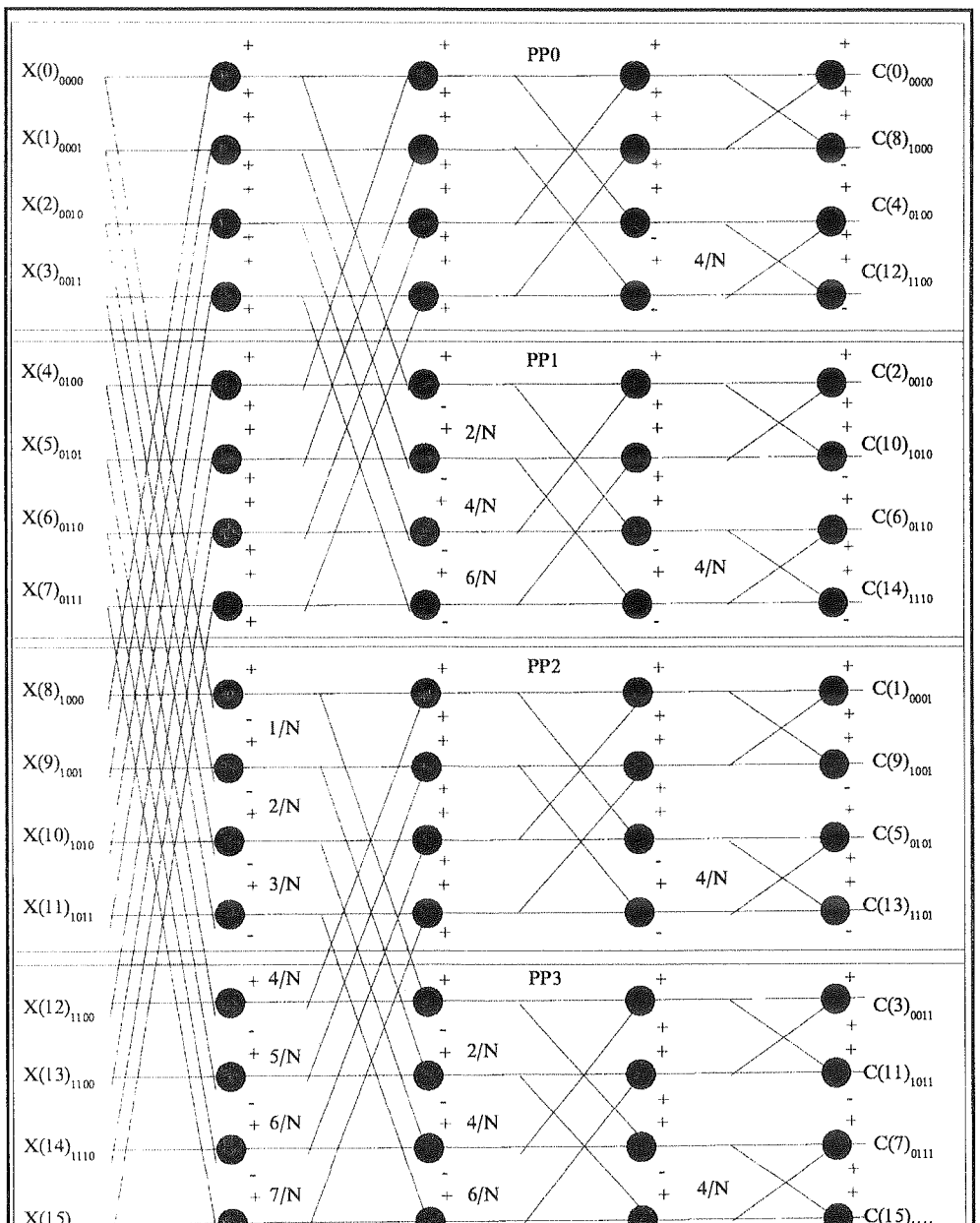


Fig. 20. Butterfly scheme implemented for four DSP processors.

9. FINAL CONCLUSIONS

The vision systems constitute the important class of computer systems, are the subject of many research works and find many practical applications [16]. The continual growth of camera resolution and the number of images delivered in one second require also very fast image processing methods [17], [18], [19], [20], [21]. The search of new multiprocessor architectures possessing great computational power is the field of many research works [22], [23], [24], [25], [26], [27], [28], [29], [30]. In the very fast image processing systems specialised processoers often are used. There are known many examples where specialised processors were implemented for the realisation of some basic image pre-processing operation such as: median filtering [31], [32], [33], [34]; convolution [35], [36]; edge detection [37]; histogram calculation [38], [39], [40] and minimum and maximum search [41]. In modern image processing systems the re-configurable FPGA structures are also popular [42], [43], [44], [45], [46], [47].

In compresion with the above mentioned image processing systems the Texas Instruments multiprocessor chip the TMS320C80 constitutes the new and totaly different solution. The TMS320C80 combines the great computation power (a feature characteristic for specialised processors) with full programmability (a feature characteristic for general-purpose processors). It is also an important fact that the processors of the TMS320C80 can be programmed directly in ANSI standard of commonly known C language. In opposite to the specialised processors or FPGA structures the programming of the TMS320C80 is relatively simple and does not require the knowledge of some special programming language. In addition some special features of the TMS320C80 architecture such as loosely coupled processor and selfconfiguring crossbar switching network, which sets automatically all the necessary data paths for inter-processor communication, allows for the implementation of different types of multiprocessor architectures. It very important fact because the architecture of multiprocessor system plays the main role in effective image data processing and is the main factor that possess the influence on the processing time in such systems [48], [49], [50], [51], [52], [53], [54], [55], [56].

REFERENCES

1. L. Macaire, J. G. Postaire: *Automated visual inspection of galvanised and painted metallic strips*. 1993 CompEuro Proceedings, Computers in Desing, Manufacturing and Production, Paris-France, 1993, pp. 8 – 15.
2. B. Mehenni, M. A. Wahab: *APRIS: Automatical pattern recognition and inspection system*. 1993 CompEuro Proceedings, Computers in Design, Manufacturing, and Production, Paris-France, 1993, pp. 23 – 26.
3. T. Thomas, M. Cattoen: *System for defect on materials*. 1993 CompEuro Proceedings, Computers in Design, Manufacturing, and Production, Paris-France, 1993, pp. 16 – 22.
4. B. Kwolek: *Komputerowe przetwarzanie obrazów w wieloprocesorowym systemie czasu rzeczywistego*. Conference Materials of „VI Konferencja Systemów Czasu Rzeczywistego", Zakopane, 1999, ss. 271 – 280.
5. R. Tadeusiewicz: *Systemy wizyjne robotów przemysłowych*. Warszawa, WNT, 1993.

6. S. Kozielski, Z. Szczerbiński: *Komputery równoległe*. Warszawa, WNT, 1994
7. B. S. Chalk: *Organizacja i architektura komputerów*. Warszawa, WNT, 1998
8. A. Skorupski: *Podstawy budowy i działania komputerów*. Warszawa, WKiŁ, 1997
9. A. S. Tanenbaum: *Rozproszone systemy operacyjne*. Warszawa, WKiŁ, 1997
10. *TMS320C80 System Lovel Synopsis*. Catalogue Material of the Texas Instruments
11. M. J. Flynn: *Very high-speed computing-systems*. Proceedings of the IEEE, 1966, Vol. 54, No. 12
12. P. P. Jonker: *Morphological image processing: architecture and VLSI desing*. Norwell MA, Kluwer 1992
13. C. M. Chang, S. L. Lu: *Designe of a static MIMD data flow processor using micropipelines*. IEEE Transactions on Very Large Scale Intergration (VLSI) Systems, Vol. 3, No 3, September 1995, pp. 370–378.
14. W. Gehrke, K. Gaedke: *Associative controlling of monolithic parallel processor architectures*. IEEE Transactions on Ciruits and Systems for Video Tehnology, Vol. 5, No. 5, October 1995, pp. 453–464.
15. E. Mirsky, A. De Hon: *MATRIX: A reconfigurable computing architectuers with configurable instruction distribution and deployable resources*. Proceedings of the IEEE Symposium on FPGAs for fast Costom Computing Machines, California 1996, pp. 157–166.
16. A. Antola, A. Avai, L. Breveglieri: *Modular desing methodologies for image processing architectures*. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 1, No. 4, Dec. 1993, pp. 408–414.
17. J. Hyneck: *Low-noise and high-speed charge detection in high-resolution CCD image sensor*. IEEE Transactions on Electron Devices, Vol. 44, No. 10, October 1997, pp. 1679–1688
18. K. Itakura, T. Nobusada, Y. Toyoda, Y. Saitou: *A2/3-in 2.0 M.-Pixel CCD Imager with an advanced M-FIT architecture capable of progressive scan*. IEEE Transactions on Electron Devices, Vol. 44, No.10. October 1997, pp. 1625–1632
19. T. Nomoto, S. Hosokai, T. Isokawa, R. Huga: *A 4-M-Pixel CMD Image Sensor with block and skip access capability*. IEEE Transactions on Electron Devices, Vol. 44, No. 10, Oct 1997, pp. 1738–1746
20. T. Ozaki, H. Ono, H. Tanaka, S. Sato, M. Nakai, T. Nishida: *A high-packing density pixel with punchthrough read-out method for an HDTV interline CCD*. IEEE Transactions on Electron Devices, Vol. 41. No 41. July 1994, pp. 1128–1135
21. D. Scheffer, B. Dierick, G. Meynants: *Random addressable 2048*2048 active pixel image sensor*. IEEE Transactions on Electron Devices, Vol. 44, No 10, October 1997, pp. 1716–1720
22. P. Bagalietto, M. Maresca, M. Migliardi, N. Zingrian: *Image processingon high- performance RISC systems*. Proceedings of the IEEE, Special Issue on Parallel Architectures for Image Processing, Vol. 84, No 7, July 1996, pp. 917–930
23. M. J. Colaitis, J. L. Jumpertz, B. Guerin, B. Cheron, F. Battini, B. Lescure, E. Gautier, J. P. Geffroy: *The implementation of P³I, a parallel architecture for video real-time processing: A case study*. Proceedings of the IEEE, Special Issue on Parallel Architectures for Image Processing, Vol. 84, No 7, July 1996, pp. 1019–1037
24. F. Distante, M. Sami, R. Stefanelli, G. Storti-Gajani: *Mapping neural nets onto a massively parallel architecture: A defect-tolerance solution*. Proceedings of the IEEE, Vol. 79, No 4, April 1991, pp. 444–460
25. R. M. Lea, J. P. Jalowiecki: *Associative massively parallel computers*. Proceedings of the IEEE, Vol. 79, No. 4, April 1991, pp. 469–479
26. H. Li, Q. F. Stout: *Reconfigurable SIMD Massively Parallel Computers*. Proceedings of the IEEE, Vol. 79, No. 4, April 1991, pp. 429–443
27. D. H. Shaefer: *The characterisation and representation of massively parallel computing structures*. Proceedings of the IEEE, Vol. 4, April 1991, pp. 461–468
28. J. P. Strong: *Computations on the massively parallel processor at the Goddar Space Filght Center*. Proceedings of the IEEE, Vol. 79, No 4, April 1991, pp. 548–558
29. S. S. Wilson: *Teching metwork connectivity using simulated annealing on a massively parallel processors*. Proceedings of the IEEE, Vol. 79, No. 4, April 1991, pp. 559–566

30. N. B. Wilding, A. S. Trew, K. A. Hawick, G. S. Pawley: *Scientific modeling with massively parallel SIMD computers.* Procedings of the IEEE, Vol. 79, No. 4, April 1991, pp. 547 – 585
31. C. T. Chen, L. G. Chen, J. H. Hsiao: *VLSI implementation of a selective median filter.* IEEE Transactions on Consumer Electron, Vol. 42, No. 1. Februaty 1996, pp. 31 – 41
32. J. C. Gealow, F. P. Hermann, L. T. Hus, C. G. Sodini: *System designe for pixel-parallel image processing.* IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 4, No. 1, March 1996, pp. 32 – 41
33. B. K. Kar, K. M. Yusuf, D. K. Pradhan: *Bit-serial generalised median filters.* Proceedings of the IEEE International Symposium on Circuits an Systems, London, 1994, pp. 5.85 – 3.88
34. J. Siu, J. Li, S. Luthi: *A real-time 2-D median based filter video signals.* IEEE Transactions on Consumer Electronics, Vol. 39, No. 2, May 1993, pp. 115 – 121.
35. D. W. Hammerstrom, D. P. Lulich: *Image processing using ane-dimensional processor arrays.* Proceedings of the IEEE, Special Issue on Parallel Architectures for Image Processing, Vol. 4, No 7, July 1996, pp. 1005 – 1018
36. P. J. Rose, B. G. Koether: *A 75 MHz CMOS digital convolver.* Proceedings of the IEEE International Conference on Computer Desing: VLSI in Computers & Processors, Massachusetts MA, 1990, pp. 311 – 314
37. M. N. Fesharaki, G. R. Hellestrand: *A real-time edge detection ASIC desing.* Proceedings of the IEEE , London, April 1994, pp. 3.81 – 3.84
38. M. Abdelguerfi, S. Khalaf, A. K. Sood: *Bit-serial parallel processing unit for the histogramming operation.* IEEE International Symposium on Ciruits ans Systems, Vol. 37, No. 7, July 1990, pp. 948 – 954
39. S. Muller: *A new programmable VLSI architecture for histogram and statisc computation in different windows.* Proceedings of the IEEE International Conference on Image Processing, Washington DC, 1995, pp. 73 – 76
40. K. Ronner, J. Kneip: *Architecture and Applications of the HiPar Video Signal Processor.* IEEE Transactions on Ciruits and Systems for Video Technology, Vol. 6, No. 1, February 1996, pp. 56 – 66
41. C. Lee, S. C. Juan, W. W. Yang: *A parallel bit-level maximum/minimum selector for didital and video signal processing.* IEEE Transactions On Circuits and Systems-II: Analog and Digital Signal Processing, Vol. 41, No. 10. October 1994, pp. 693 – 695
42. H. R. Arabnia, J. W. Smith: *A reconfigurable parallel imaging systems.* Proceedings of the IASTED & IEEE International Conference: Signal and Image Processing — SIP-95 Las Vegas NV 1995, Anaheim-Calgary-Zurich IASTED Acta Press, pp. 21 – 23
43. P. J. Bekkes, J. J. Plessis, B. L. Hutchings: *Maxing fixed and reconfigurable logic array processing.* Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, California 1996, pp. 118 – 127
44. B. Gunther, G. Milne, L. Narasimhan: *Assessing document relevance with run-time reconfigurable machines.* Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, California 1996, pp.10 – 17
45. R. Miller, V. K. Prasanna - Kumar, D. I. Reisis, Q. F. Stout: *Image Computation on reconfigurable VLSI arrays.* IEEE Transactions 1988, pp. 925 – 930
46. J. Villasenor, B. Schoner, K. N. Chia, C. Zapata: *Configurable computing solutins for automatic target recognition.* Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, California 1996, pp. 70 – 79
47. T. Yamauchi, S. Nakaya, N. Kajihara: *SOP: A reconfigurable massivly parrallel systems and its control-data-flow based computing method.* Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, California, 1996, pp. 148 – 156
48. W. E. Alexander, D. S. Reeves, C. S. Gloster: *Parallel image processing with the block data parallel architecture.* Proceedings of the IEEE, Special Issue on Parallel Architectue for Image Processing, Vol. 84, No. 7, July 1996, pp. 947 – 968
49. N. Alexandridis: *Desing of microprocessor-based systems.* New Jersey, Prentice Hall, 1993
50. M. J. Flynn: *Very high-speed computong-systems.* Procedings of the IEEE, 1996, Vol. 54, No. 12

51. M. H a m d i: *Modified mesh-connected computers for image processing applications.* IEEE Transactions 1995, pp. 139–145
52. K. H w a n g: *Advanced computer architecture.* New York, Mc Graw-Hill 1993
53. R. Y. K a i n: *Advanced computer architecture.* New Jersey, Pratice Hall 1996
54. D. M i l f o r d: *Architecture for computer vision.* Applied mathematics and computer scuence, vol. 3, No. 1, 1993, pp. 135–140
55. R. L. S o h i e, W. C h e n: *Implementation of fast Fourier transforms on Motorola's digital signal processors.* Motorola Inc., 1993
56. A. Ś l u z e k: *Komputerowa analiza obrazów.* Warszawa, Wydawnictow Politechniki Warszawskiej, 1991

This work was supported by KBN grant 8T11 03914 „The analysis and desing of the real-time systems of the different distribution grade"

M. GAJER

IMPLEMENTACJA ALGORYTMÓW PRZETWARZANIA OBRAZÓW DLA SYSTEMU WIELOPROCESOROWEGO

S t r e s z c z e n i e

Aplikacje z dziedziny przetwarzania i rozpoznawania obrazów cyforwych, charakteryzują się dużym zapotrzebowaniem na moc obliczeniową procesorów. Związane jest to z faktem, że te same operacje, takie jak np. filtracja, wyostrzenie obrazu, detekcja krewędzi, wyrównywanie histogramu, czy też binaryzacja obrazu wykonywane muszą być dla każdego z pikseli z osobna, co przy rozmiarach obrazów rzędu kilkuset tysięcy pikseli powoduje znaczne wydłużenie czasu obliczeń. Jeżeli jeszcze obrazy te mają być przetwarzane, analizowane i rozpoznawane w czasie rzeczywistym, (jako przykład można tutaj podać system wizyjny robota przemysłowego, system radarowy sterujący lotem samolotu itp.) wówczas zapewnienie dostatecznej mocy obliczeniowej staje się jeszcze ważniejszym problemem. Dostępne obecnie na rynku procesory nie zawsze są w stanie dostarczyć odpowiedniej mocy obliczeniowej, zatem konstruktorzy takich systemów coraz częściej sięgają w stronę rozwiązań wieloprocesorowych, w celu przyspieszenia obliczeń.

Naprzeciw zapotrzebowaniom na moc obliczeniową ze strony aplikacji związanych z przetwarzaniem i analizą obrazów wyszła firma Texas Instruments, wypuszczając na rynek w 1995 roku pierwszy i do tej pory jedyny system wieloprocesorowy wykonany w postaci pojedynczego układu scalonego.

Isnieje wiele różnych możliwych sposobów zrównoleglenia obliczeń, związanych z przetwarzaniem obrazów. Wybór odpowiedniej konfiguracji systemu wieloprocesorowego jest niezykle istotną sprawą, poniważ prawidłowy dobór architektury systemu posiada decydujący wpływ na wartość współczynnika przyśpieszenia obliczeń, w przypadku operacji przetwarzania obrazów, polegających na wykonaniu operacji splotu obrazu z maską filtru, można uzyskać największe wartości współczynnika przyśpieszenią obliczeń, w przypadku zastosowania architektury typu SMD. W tym przypadku obraz dzielony jest na tyle równych części, ile procesorów posiada system, a następnie wszystkie procesory wykonują te same obliczenia dla różnych zestawów danych. Z kolei operacje przetwarzania obrazów, składające się z wielu etapów, mogą zostać najefektywniej zrównoleglone poprzez zastosowanie wieloprocesorowych struktur potokowych, gdzie każdy z etapów przetwarzania obrazu wykonywany jest przez odrębną stację potoku.

Słowa kluczowe: Analiza i przetwarzanie obrazów, systemy wieloprocesorowe, obliczenia równoległe, architektury systemów równoległych

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2000, 46, z. 2, ss. 273–274

SPIS TREŚCI

CONSTENTS

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.

Wymagania podstawowe: Artykuły należy nadsyłać w maszynopisie pisanym jednostronnie lub na wyraźnym czarno-białym wydrkuku komputerowym, w 2 egzemplarzach, w języku polskim lub angielskim wybranym przez autora. Tekst artykułu musi być poprzedzony tytułem pracy, imieniem i nazwiskiem Autora wraz z podaniem miejsca jego pracy. 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 mięc około 35 mm. Przy podziale pracy na rozdziały i podrozdziały cyfrowe ich oznaczenia nie powinny być większe niż III stopnia (np. 4.1.1).

Sposób pisania tablic: Tablice powinny być pisane na oddzielnych stronach. Tytuły rubryk pionowych i poziomych powinny być napisane małymi literami z podwójnym odstępem między wierszami. Przypisy (notki) dotyczące tablic należy pisać bezpośrednio pod tablicą. Tablice należy numerować kolejno liczbami arabskimi, u góry każdej tablicy podać tytuł. Tablice umieścić na końcu maszynopisu. Przyjmowane są tablice algorytmów i programy na wydrukach komputerowych. W tym przypadku zachowany jest ich oryginalny układ.

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 z wytycznymi IEE (International Electronical Commision) oraz ISO (International 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: F. Valdoni: A new milimetre wave satelite. E.T.T. 1990, vol. 2, no 5, p. 553

nieperiodycznej: K. Andersen: A resource allocation framework. XVI International Symposium, Stockholm (Sweden), may 1991, paper A 2.4

książki: Y.P. Tvidis: Operation and modeling of the MOS transistors. New York, Mc Graw-Hill, 1987, pp. 141 – 148

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. 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. Spis podpisów pod rysunki i fotografie powinien być umieszczony na oddzielnej stronie.

Streszczenia: Do każdego artykułu musi być dołączone obszerne — do 60 wierszy — streszczenie z tytułem artykułu. Streszczenia (Summary) muszą być w języku polskim i angielskim. Streszczenie powinno wyjaśniać główny cel pracy, wskazywać korzyści i ograniczenia, możliwe zastosowania i zalecenia dla dalszego rozwoju danej gałęzi techniki. Pod streszczeniami powinny być podane słowa kluczowe.

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 Redkacji i 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.

Redkacja prosi Autorów o powiadomienie ją o zmianie miejsca pracy i adresu prywatnego.

INFORMATION FOR THE AUTORS OF K.E.T.

The KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI (K.E.T.) — ELECTRONICS AND TELECOMMUNICATION QUARTERLY is a journal of the Committee for Electronics and Telecommunication of the Polish Academy of Sciences and is a direct continuation for 46 years of the quarterly ROZPRAWY ELEKTROTECHNICZNE, that has been published also under the auspices of the Polish Academy of Sciences.

Its pages contain scientific articles concerned with theory and application of electronics, telecommunication, microelectronics, optoelectronics, radioengineering and medical electronics.

Accordingly, its pages contain scientific articles, that are original works both contributory and concerned with comparative analyses of methods or systems synthetic reviews of a given field, state-of-art discussions of a given field, and presentations of developments ina given technology.

Regular papers are accepted for publication in Polish or English, but title, the estensive summary and figure caption must be bilingual. Bilingual summary must shown the significance of the results shown in the paper, emphasizing limitations and advantages, possible applications and recommendations for further works.

K.E.T. is meant for specialists and students with above subjects.

FORMAL PREPARATION OF MANUSCRIPT

Texts

Texts to be submitted to K.E.T. may be typed on one side of the paper only with double spacing between the lines and a 4 cm margin or using of wordprocessing with text formatters based on MS DOS, APPLE/MAC such as Microsoft word. In any case the text formatter must be specified on the Author/s, their Affiliations and relevant addresses.

All text pages should be numbered. Regular papers should have a length of no more than 30 pages.

Printing types and symbols

For letteral symbols, units and graphical symbols, the recommendation of IEE (International Electrotechnical Commission) and ISO (International Orgnisation of Standarisation) must be followed.

- In case of manuscripts conventionally typed (i.e. without wordprocessing), letters to be printed in italic should be underlined; letters to be printed in bold should be double interlined.
- Formulae and footnotes should be numbered in accordance with the quotation order followed in the text.
- The serial number of each formulae should be written at right side, between round brackets.
- The serial number of footnotes should be written between round brackets and upper.
- Mathematical detials, anciliary to the main discussion of the paper may be included in one or more appendixes, that are printed after conclusions and before references.

In manuscript containing lot of symbols, a glossary may be included after introduction.

References

References are usually gathered at the end of the text and numbered according to the order of quotation in the text. The serial numbers should be written without square brackets (i.e. 1, 2...)

Illustrations

The originals of illustrations may be either drawings or photographs (black or colour). Each copy of paper must include two complete sets of illustrations.

Subscription for external subscribers:

The promotional subscription price in 2000 is $ 90 including postage for institutions. Subscriptions should be sent to the publisher, Polish Scientific Publishers PWN LTd., Journal Division, Miodowa 10, 00-250 Warsaw, POLAND, fax. (48) (22) 826 09 50, (48) (22) 826 71 63, with a copy by fast to Electronics and Telecommunications Quarterly 00-665 Warsaw, ul. Nowowiejska 15/19, 470. Subscription is occupated after showing a cheque for transfer documents. Our bank account is as follows: Bank account: PBK VIII O/Warszawa nr 11101037-1052-2700-1-43. At subscripber's request this journal will be air mailed at additional postage to 50% dross proce to European countries and 65% overseas.

Subscription orders available through the local press distriution or through the Foreign Trade Enterprise ARS Polona, 00-068 Warszawa, Krakowskie Przedmieście 7, Poland.

Ruch S.A. fulfills foreign customers orders, starting from any issue in the calendary year: (48) (22) 620 120 39, fax: (48) (22) 620 17 62.