Adres Redakcji
Wrocławska Wyższa Szkoła Informatyki Stosowanej, Wydział Informatyki, ul. Wejherowska 28, 54-239 Wrocław, tel. (71) 788-94-25, fax (71) 799-19-36
e-mail: wydawnictwo@horyzont.eu

Skład i łamanie:
Marcin Radzewicz

Druk:
Wrocławska Wyższa Szkoła Informatyki Stosowanej

Spis treści · Contents

Wstęp *Preface*

Drodzy Czytelnicy,

W imieniu redakcji i swoim mam przyjemność przekazać na Państwa ręce pierwszy numer czasopisma *Informatyka* ukazującego się w ramach *Biuletynu Naukowego Wrocławskiej Szkoły Informatyki Stosowanej* we Wrocławiu. Pragnąc poszerzyć horyzonty naszej uczelni i dać możliwość rozwoju zarówno kadrze dydaktycznej, jak i studentom, narodził się pomysł utworzenia biuletynu naukowego poświęconego zagadnieniom związanym z informatyką.

Informatyka jest jednym z najszybciej rozwijających się obszarów nauki, a zarazem tym, który nieodłącznie towarzyszy nam na co dzień w różnych aspektach życia. Różnorodność zastosowań nauk informatycznych doprowadziła do powstania wielu jej specjalności, począwszy od algorytmiki i elektroniki, poprzez programowanie i zarządzanie oprogramowaniem, aż po zastosowania w innych dyscyplinach nauki takich jak medycyna, mechanika, czy fizyka. Możliwość przeprowadzania skomplikowanych symulacji komputerowych, a także tworzenia inteligentnych systemów, znacznie przyczyniła się do rozwoju innych gałęzi nauki.

Dzisiaj pragniemy zaprezentować Państwu pierwszy numer *Informatyki*, na łamach której publikowane będą prace z różnych dziedzin informatyki. Tematyka poruszana w artykułach naszego czasopisma obrazuje szerokie zastosowania tychże dziedzin. Naszym celem jest harmonijny rozwój biuletynu wraz z rozwojem informatyki oraz zachowanie wysokiego poziomu prezentowanych prac. Jako redaktor naczelny mam nadzieję, iż publikowane przez niniejsze czasopismo artykuły dadzą naszym Czytelnikom możliwość pogłębiania wiedzy i staną się inspiracją do dalszego rozwoju.

Życzę przyjemnej lektury

Łukasz Jeleń

Redaktor naczelny

Prosta, efektywna kwadratura adaptacyjna w języku C

A simple and effective adaptive quadrature in C

Paweł Keller

Wrocławska Wyższa Szkoła
Informatyki Stosowanej
ul. Wejherowska 28, 54-239 Wrocław

Treść. Prezentujemy dwa proste, a jednocześnie bardzo skuteczne algorytmy adaptacyjne przybliżania wartości całki oznaczonej funkcji rzeczywistej. Proponowane algorytmy działają w oparciu o zasadę *dziel i zwyciężaj*, i wykorzystują znane kwadratury wysokiego rzędu.

Słowa kluczowe: kwadratura, kwadratura adaptacyjna, całkowanie numeryczne, całka oznaczona.

Abstract. We present two simple and very effective adaptive algorithms for approximating a definite integral of a real function. The proposed methods are based on the *divide and conquer* rule and use well known high order quadrature rules.

Keywords: quadrature, adaptive quadrature, numerical integration, definite integral.

1 Wprowadzenie

Konieczność obliczenia liczbowej wartości całki oznaczonej danej funkcji ciągłej pojawia się w niezliczonej ilości zagadnień naukowych i technicznych. W sytuacji, kiedy całka nieoznaczona pewnej funkcji nie daje się wyrazić za pomocą funkcji elementarnych, zwykle jedynym sposobem na wyznaczenie całki oznaczonej jest przybliżenie jej wartości jedną z metod całkowania numerycznego. Metody takie nazywamy *kwadraturami*. W niniejszej pracy zajmować się będziemy problemem wyznaczenia wartości całki

$$\int_a^b f(x)\,dx\,, \tag{1.1}$$

przy założeniu, że potrafimy obliczyć wartość funkcji podcałkowej f w dowolnym punkcie przedziału $[a,b]$. Zakładamy przy tym, że funkcja podcałkowa nie jest z góry znana. Oznacza to, że nie możemy dopasować metody całkowania do konkretnej funkcji występującej w (1.1). Najlepszym rozwiązaniem w takiej sytuacji wydają się być kwadratury adaptacyjne, które z założenia powinny poprawnie obliczać wartość całki oznaczonej dla możliwie najszerszej klasy funkcji. Ideę działania i konstrukcji kwadratur adaptacyjnych opisujemy dokładniej w rozdziałach 2 i 3.

Kiedy korzystamy z zaawansowanego systemu obliczeń matematycznych, jak *Matlab*, *Maple* czy *Matematica*, możemy skorzystać z wbudowanych procedur całkujących. Jeśli jednak zachodzi potrzeba obliczenia całki typu (1.1) w programie pisanym w jednym z języków niższego poziomu (w C, Javie lub podobnym) musimy albo samodzielnie zaprogramować odpowiedni algorytm albo znaleźć gotową procedurę napisaną w danym języku programowania, wierząc w jej niezawodność.

Klasyką wśród publikacji naukowych dotyczących kwadratur adaptacyjnych jest praca Gandera i Gautchiego [4] z 2000 roku. Do ciekawszych pozycji należy również wcześniejsza propozycja [3], a także całkiem nowa praca Shampine [6] z 2007 roku, na podstawie której powstała nowa wektorowa kwadratura adaptacyjna dostępna od trzech lat w systemie obliczeń matematycznych *MatLab*.

Przeglądając natomiast sieć WWW w poszukiwaniu stron zawierających hasło „*kwadratura adaptacyjna w C*" lub jego angielski odpowiednik, udało się autorom łatwo odnaleźć cztery odnośniki ([8], [9], [10] i [11]) do witryn oferujących biblioteki lub procedury pozwalające obliczać w języku C całki postaci (1.1). Biblioteka [11] („NAG Library") jest płatna i nie będziemy z niej korzystać podczas testów. Procedura oferowana na stronie [8] potrzebowała natomiast aż 11 sekund aby przybliżyć wartość całki funkcji e^x na przedziale [-1,1] z błędem bezwzględnym nieprzekraczającym 10^{-10}, i dlatego również nie będzie uwzględniana podczas testów (proponowana w tej pracy kwadratura oblicza tę całkę, na tym samym komputerze, w czasie krótszym niż 10 mikrosekund, z błędem mniejszym niż 10^{-15}). Metody oferowane na stronach [9] i [10] działają poprawnie, jednak nie pozwalają zadać dokładności obliczanego przybliżenia, ani nie podają oszacowania błędu obliczonej wartości całki, co jest sporym mankamentem w wypadku kwadratur adaptacyjnych. Będą jednak wykorzystane w testach, aby można było lepiej ocenić skuteczność i efektywność proponowanych przez nas algorytmów.

W pracy prezentujemy dwa zbliżone do siebie adaptacyjne algorytmy przybliżania wartości całki (1.1). Naszym celem było zaproponowanie kwadratury prostej w swej konstrukcji (aby można ją było bardzo łatwo zaimplementować w dowolnym języku programowania), a jednocześnie szybkiej i dokładnej. Testy numeryczne przedstawione w rozdziale 4

pokazują, że cel został osiągnięty. Wcześniej, w rozdziałach 2 i 3 opiszemy dokładnie zasadę działania proponowanych kwadratur adaptacyjnych.

W rozdziale 5 podajemy, zapisane w języku C (standard ISO/IEC 9899), kody źródłowe procedur będących implementacją algorytmu opisanego w niniejszej pracy.

2 Wybór kwadratury podstawowej i oszacowanie błędu przybliżenia

Idea działania kwadratury adaptacyjnej jest bardzo prosta. Wartość całki (1.1) nie jest obliczana *od razu* na całym przedziale $[a, b]$, ale przedział ten dzielony jest na szereg podprzedziałów i ostateczny wynik jest obliczany jako suma wartości całek na każdym z nich. Co istotne, podprzedziały te nie są jednakowej długości. W miejscach, w których funkcja podcałkowa zachowuje się regularnie są one dłuższe, a w obszarach silnej zmienności funkcji znacznie krótsze. Ponieważ, jak zaznaczono we wstępie, funkcja f w (1.1) nie jest z góry znana, podział odcinka $[a, b]$ musi być tworzony dynamicznie w trakcie obliczania wartości całki.

W tej pracy rozważać będziemy jedynie schemat doboru długości podprzedziałów oparty na zasadzie połowienia. Ogólnie, sposób działania kwadratury adaptacyjnej można opisać następująco:

Algorytm 1 *Schemat adaptacyjnego algorytmu przybliżania wartości całki $\int_a^b f(x)dx$ z błędem bezwzględnym nie przekraczającym δ.*

Krok 1. Wyznacz *pewnym sposobem* przybliżenie I całki funkcji f na przedziale $[a, b]$ oraz oszacowanie ε błędu bezwzględnego tego przybliżenia.

Krok 2. Jeśli $\varepsilon < \delta$, zaakceptuj I jako dobre przybliżenie całki i zakończ.

Krok 3. Wyznacz środek przedziału: $m = (a + b)/2$.

Krok 4. Oblicz rekurencyjnie przybliżenie I_1 całki $\int_a^m f(x)dx$ z błędem bezwzględnym nie przekraczającym $c\,\delta$, dla pewnej ustalonej stałej $\frac{1}{2} \le c \le 1$.

Krok 5. Oblicz rekurencyjnie przybliżenie I_2 całki $\int_m^b f(x)dx$ z błędem bezwzględnym nieprzekraczającym $c\,\delta$.

Krok 6. Przyjmij $I = I_1 + I_2$ i zakończ.

Łatwo zauważyć, że sumując oszacowania błędów bezwzględnych obliczonych przybliżeń całek na wyznaczonych w algorytmie podprzedziałach otrzymamy oszacowanie błędu całki na całym przedziale $[a, b]$.

Pozostają jednak wciąż dwie niewyjaśnione kwestie. Jak w kroku 1 wyznaczyć przybliżenie I oraz jak wyznaczyć jego wiarygodne oszacowanie ε. Zanim to opiszemy zajmiemy się jeszcze problemem żądanej tolerancji błędu w krokach 4 i 5.

Jest rzeczą intuicyjną, że skoro przedział całkowania jest dzielony na dwie równe części, to chcemy aby błędy przybliżeń całki na każdej połowie przedziału nie przekraczały $\frac{1}{2}\delta$. Wtedy błąd na całym przedziale będzie na pewno nie większy niż δ. W [7, rozdz. 5.2] uzasadniono jednak, że takie podejście jest nazbyt ostrożne i niepotrzebnie wydłuża obliczenia. Zwykle w krokach 4 i 5 Algorytmu 1 można przyjąć $c = 1$, jednak dla „trudnych" funkcji zbyt często zdarza się wtedy, że otrzymany błąd przybliżenia całki na całym przedziale jest nazbyt duży. Na podstawie wielu przeprowadzonych doświadczeń, w zaproponowanych w tej pracy metodach adaptacyjnych przyjmujemy $c = 0.8125$.

Powróćmy teraz do zagadnienia wyznaczenia przybliżenia całki w kroku 1 Algorytmu 1. Użytą w tym celu kwadraturę nazywać będziemy *kwadraturą podstawową*. W naszej pracy rozważać będziemy jedynie kwadratury liniowe postaci

$$Q(f, a, b) = \sum_{k=0}^{n} w_k f(x_k) \approx \int_a^b f(x)\,dx\,, \qquad (2.1)$$

gdzie ustalone wartości $a \le x_0 < x_1 < \cdots < x_n \le b$ nazywamy *węzłami* kwadratury, a wielkości $w_0, w_1, \ldots, w_n$ *wagami* lub *współczynnikami* kwadratury. Spośród kwadratur postaci (2.1) nie da się wybrać jednej, która gwarantowałaby najmniejszy błąd przybliżenia dla wszystkich całkowalnych funkcji f. Dlatego często stosuje się inną miarę jakości danej kwadratury.

Definicja 1 Mówimy, że kwadratura (2.1) jest *rzędu* r, jeśli jest ona dokładna dla każdego wielomianu stopnia niższego niż r, ale już nie dla każdego wielomianu stopnia r.

Ponieważ każdą funkcję ciągłą można dowolnie dobrze przybliżyć na odcinku domkniętym za pomocą wielomianu, można przypuszczać, że im wyższy rząd kwadratury, tym powinna ona średnio lepiej przybliżać wartość całki. W naszej metodzie wykorzystamy kwadratury możliwie najwyższych rzędów.

Omówimy teraz kwestię oszacowania ε błędu przybliżenia całki w kroku 1 Algorytmu 1, czyli oszacowania wielkości

$$\left| \int_a^b f(x)\,dx - Q(f, a, b) \right|.$$

Najczęstszym rozwiązaniem tego zagadnienia jest zastosowanie pary kwadratur

$$\begin{aligned} Q_1(f,a,b) &= \sum_{k=0}^{n} w_k f(x_k), \\ Q_2(f,a,b) &= \sum_{k=0}^{s} v_k f(y_k), \end{aligned} \tag{2.2}$$

gdzie kwadratura Q_2 powinna być teoretycznie znacznie dokładniejsza niż kwadratura Q_1. Za przybliżenie wartości całki przyjmuje się wartość kwadratury Q_2, natomiast błąd bezwzględny przybliżenia szacuje się w oparciu o wielkość modułu różnicy wartości obu kwadratur. Często po prostu przyjmuje się

$$\varepsilon = |Q_1(f,a,b) - Q_2(f,a,b)|, \tag{2.3}$$

choć nie jest to regułą.

W praktyce ważne jest też, aby $\{x_0, x_1, \ldots, x_n\} \subset \{y_0, y_1, \ldots, y_s\}$. Wtedy żadna z obliczonych wartości $f(x_k)$ $(k = 0, 1, \ldots, n)$ się nie marnuje.

Najbardziej popularnymi w profesjonalnych algorytmach adaptacyjnych obliczania całek parami kwadratur postaci (2.2) są tak zwane pary Gaussa-Kronroda. Kwadratura Q_1 jest kwadraturą Gaussa-Legendre'a, czyli kwadraturą mającą możliwie najwyższy rząd przy ustalonej liczbie węzłów. Kwadratura Q_2 jest tak zwanym rozszerzeniem Kronroda, czyli kwadraturą o najwyższym rzędzie spośród kwadratur opartych na $2n + 3$ węzłach zawierających wszystkie węzły kwadratury Q_1 (więcej informacji na temat kwadratur Gaussa-Kronroda można znaleźć w [2, rozdz. 2.7.1.1].

W proponowanej w naszej pracy metodzie zastosujemy inne podejście. Skorzystamy mianowicie z pojedynczej kwadratury podstawowej. Dzięki temu mamy możliwość zastosowania w schemacie adaptacyjnym praktycznie dowolnej kwadratury postaci (2.1). Do oszacowania błędu i przybliżenia wartości całki wykorzystamy natomiast kwadratury utworzone następująco:

$$\begin{aligned} Q_1(f,a,b) &= Q(f,a,b), \\ Q_2(f,a,b) &= Q(f,a,m) + Q(f,m,b), \end{aligned} \tag{2.4}$$

gdzie $m = (a+b)/2$, a Q jest wybraną kwadratura podstawową. Jeśli kwadratura Q w (2.1) ma wysoki rząd, to (porównaj [7, rozdz. 5.1]) jej błąd jest proporcjonalny do $(b-a)^p$, gdzie $p \gg 1$ (przy pewnych założeniach dotyczących funkcji f). Zdefiniowana w (2.4) kwadratura Q_2 powinna być zatem znacznie dokładniejsza niż kwadratura Q_1. Oszacowanie błędu będziemy obliczać zgodnie z (2.3).

Zauważmy, że przy takim podejściu w każdym rekurencyjnym kroku Algorytmu 1 oszczędzamy czas potrzebny na wyznaczenie wartości kwadratury Q_1, ponieważ odpowiednie wartości były już wyznaczone wcześniej dla każdej z obu połówek przedziału.

W naszej pracy proponujemy dwie metody adaptacyjne obliczania całki (1.1). Pierwsza z nich korzysta z kwadratury podstawowej Gaussa-Legendre'a. Przypomnijmy, że jest to kwadratura postaci (2.1) mająca możliwie najwyższy rząd (równy $2n + 2$) przy zadanej liczbie węzłów. Dodatkowe informacje na temat kwadratur Gaussa-Legendre'a można znaleźć np. w [1, rozdz. 5.3.2], [2, rozdz. 2.7] lub [5, rozdz. 7.3]. W drugiej, bliźniaczej metodzie kwadraturą podstawową jest kwadratura Lobatto. Jest to kwadratura postaci (2.1) mająca możliwie najwyższy rząd przy ustalonej liczbie węzłów i dodatkowych warunkach $x_0 = a$ i $x_n = b$ (więcej informacji na temat kwadratur Lobatto można znaleźć w [1, rozdz. 5.3.3] lub [2, rozdz. 2.7.1])

Kwadratury postaci (2.1) dla których $x_0 > a$ i $x_n < b$ nazywane są często kwadraturami *otwartymi*. Jeśli żaden z dwóch powyższych warunków nie jest spełniony, mówimy o kwadraturze *zamkniętej*. Ogólnie uważa się, że kwadratury otwarte mają lepsze własności aproksymacyjne, jednak nie powinny być one używane w algorytmach adaptacyjnych do przybliżania całek funkcji nieciągłych lub mających nieciągłą pierwszą pochodną. Rozważmy na przykład funkcję $f(x) = |x - d|$, gdzie d jest takie, że $d < x_0$, $d < y_0$ oraz $d > a$. W takiej sytuacji obie kwadratury Q_1 i Q_2 w (2.2) będą miały identyczną wartość, równą całce funkcji $g(x) = x - d$ (zakładamy, że kwadratury te są wysokiego rzędu, więc są dokładne dla funkcji liniowej). Zatem obliczone oszacowanie błędu wyniesie 0 i algorytm adaptacyjny się zakończy. W rzeczywistości natomiast błąd będzie równy $(d-a)^2$.

Mankamentu tego nie mają kwadratury zamknięte. Tych jednak nie można zastosować do przybliżania całek funkcji mających osobliwości na końcach przedziału lub w dowolnym punkcie podziału. Problem pojawia się również wtedy, gdy w takim punkcie wartość funkcji podcałkowej jest symbolem nieoznaczonym typu 0/0 (jak na przykład dla funkcji $x^{-1}\sin x$ w punkcie 0). Z tego też powodu w naszej pracy przedstawiamy dwie propozycje kwadratur adaptacyjnych.

Na zakończenie tego rozdziału pozostało jeszcze do ustalenia, ile węzłów mają mieć zastosowane w proponowanych algorytmach kwadratury podstawowe. Jeśli liczba węzłów będzie zbyt mała, kwadratura podstawowa nie będzie dobrze przybliżać całki i cały algorytm może działać za wolno. Podobnie, w wypadku za dużej liczby węzłów, łączna liczba wywołań funkcji podcałkowej może być zbyt duża, co też obniży efektywność algorytmu. Innymi słowy, kwadratura podstawowa nie powinna być *za dokładna*, bo wtedy część obliczeń będzie zmarnowana (niemal taki sam rezultat można było osiągnąć w danej

arytmetyce używając mniejszej liczby węzłów). Rozsądna liczba węzłów, w zależności od funkcji podcałkowej, to od 8 do 40. Jako, że nie wiemy jakich funkcji całki ma nasz algorytm obliczać, w obu zaproponowanych przez nas kwadraturach adaptacyjnych korzystamy z rozwiązania pośredniego, 18. punktowych kwadratur podstawowych.

Jak łatwo sprawdzić, jeśli

$$Q(f,0,1) \;=\; \sum_{k=0}^{n} w_k f(x_k)\,,$$

to analogiczna kwadratura dla dowolnego przedziału $[a,b]$ wyraża się wzorem

$$Q(f,a,b) \;=\; (b-a)\sum_{k=0}^{n} w_k f((b-a)x_k + a)\,. \quad (2.5)$$

W programie realizującym algorytm adaptacyjny obliczania całki (1.1) należy zatem pamiętać zestaw węzłów i współczynników kwadratury podstawowej dla jednego wybranego przedziału. Ze względu na prostą postać wzoru (2.5) jest to zwykle przedział $[0,1]$.

3 Oszacowania błędów zaokrągleń, wcześniejsze zakończenie obliczeń

Schemat adaptacyjny opisany w poprzednim rozdziale nie nadaje się jeszcze do zastosowań praktycznych. Wystarczy, że użytkownik zada tolerancję $\delta = 0$ i algorytm może nigdy się nie zakończyć. Wprowadzenie dolnego ograniczenia na wartość tolerancji błędu bezwzględnego niewiele pomoże, ponieważ zależy to w istotny sposób od postaci funkcji podcałkowej f. Jeśli na przykład $f(x) = 10^{10}e^x$, to najmniejszy błąd bezwzględny jakiego możemy się spodziewać stosując arytmetykę podwójnej precyzji wynosi ok. 10^{-6}. Wynika z tego, że dolne ograniczenie na wartość parametru tolerancji błędu powinno być w razie potrzeby modyfikowane podczas obliczeń.

W związku z powyższym, w proponowanych algorytmach, w trakcie wyznaczania wartości kwadratury podstawowej (2.1) obliczamy dodatkowo wielkość

$$S = \frac{1}{n+1}\sum_{k=0}^{n} |f(x_k)|$$

i modyfikujemy wartość tolerancji następująco:

$$\delta \;:=\; \max\{\delta,\, \epsilon_{mach} S\}\,,$$

gdzie ϵ_{mach} to tak zwany *epsilon maszynowy*, czyli najmniejsza wartość, dla której liczby $z_1 = 1$ i $z_2 = 1 + \epsilon_{mach}$ są różne w danej arytmetyce zmiennoprzecinkowej. Dodatkowo, wyznaczamy oszacowanie bezwzględnych błędów zaokrągleń obliczania kwadratury podstawowej:

$$\sigma := (b-a)\,S\,.$$

Oszacowania błędów zaokrągleń są sumowane i na zakończenie algorytmu dodawane do końcowego oszacowania błędu bezwzględnego obliczonego przybliżenia całki.

Eksperymenty pokazały, że w wypadku kwadratury podstawowej Gaussa-Legendre'a oraz stromo nachylonych i trudnych do całkowania funkcji obliczone oszacowanie błędu może być czasami zbyt małe. Dlatego, ostatecznie, końcowe oszacowanie błędu bezwzględnego wyznaczamy następująco:

$$\bar{\varepsilon} \;=\; \varepsilon(1 + CL) + D\,\frac{\epsilon_{mach}|q_2 - q_1|}{2(b-a)} + \bar{\sigma}\,,$$

gdzie ε jest obliczane jak w (2.3), L jest liczbą rekurencyjnych, zagnieżdżonych podziałów wyjściowego przedziału, $q_1 = Q(f,a,m)$, $q_2 = Q(f,m,b)$ (porównaj (2.4)), $\bar{\sigma}$ jest sumą oszacowań błędów zaokrągleń powstałych podczas obliczania wartości q_1 i q_2, a C i D są pewnymi stałymi. W wypadku kwadratur podstawowych Gaussa-Legendre'a i arytmetyki podwójnej precyzji przyjmujemy $C = \frac{3}{2\cdot 40}$ i $D = 1$ (dla arytmetyki pojedynczej precyzji przyjmiemy $C = \frac{3}{2\cdot 20}$). Stałe te wyznaczone zostały na podstawie wielu doświadczeń. W wypadku kwadratur podstawowych Lobatto kładziemy $C = 0$ i $D = 0$. Decyzję o kontynuowaniu algorytmu adaptacyjnego (krok 2 Algorytmu 1) podejmujemy jednak tylko na podstawie wartości ε z (2.3).

Aby uniknąć zbyt długich obliczeń w wypadku ekstremalnie skomplikowanych funkcji, w praktycznej realizacji Algorytmu 1, w kroku 2 przerywamy obliczenia także jeśli: liczba rekurencyjnych podziałów początkowego przedziału jest większa lub równa 40 (20 dla pojedynczej precyzji), długość powstałego po podziale podprzedziału jest mniejsza niż $250\epsilon_{mach}$ ($250 \approx 15\,n$), liczba wywołań funkcji podcałkowej przekroczy $2{\cdot}10^7$.

W rozdziale 5 prezentujemy procedury napisane w języku C, będące implementacją opisanej wyżej metody adaptacyjnego przybliżania całki (1.1), z wykorzystaniem kwadratury Gaussa-Legendre'a jako kwadratury podstawowej. W przedstawionych procedurach umożliwiamy dodatkowo zadanie minimalnej liczby rekurencyjnych podziałów przedziału początkowego. Zwiększenie tego parametru może być przydatne w wypadku funkcji, które zmieniają się mocno na bardzo wąskim odcinku przedziału początkowego (przykładem takim może być funkcja $f(x) =$

$e^{-(10000x)^2}$, która poza odcinkiem $[-0.0006, 0.0006]$ przyjmuje wartości praktycznie równe 0).

4 Testy numeryczne

W tym rozdziale przedstawimy wyniki eksperymentów numerycznych otrzymane podczas obliczania całek

$$\int_{-1}^{1} f_i(x)\,dx$$

dla zestawu następujących funkcji testowych:

$f_1(x) = x\sin(3x)$,

$f_2(x) = (x-\frac{1}{2})^2\sin(13x) + 20e^{-(10x)^2}$,

$f_3(x) = (1.000001 + x)^{-1}$,

$f_4(x) = \sqrt{2+\cos(100x)}$,

$f_5(x) = (1+x)\sin(\frac{1}{1+x})$,

$f_6(x) = 1000\,(1+x)\sin(\frac{1}{1+x})$,

$f_7(x) = e^{\sqrt{|5x|^3}}$,

$f_8(x) = \log(1+x)\sqrt{\frac{2+x}{1-x}}$,

$f_9(x) = \log(\cos(30x)^2)$,

$f_{10}(x) = |\cos(20.001\,\pi x)|$.

Wszystkie całki obliczano na tym samym przedziale, aby zmniejszyć ilość informacji zawartych w tabelach z wynikami. Obliczenia wykonano na komputerze z procesorem Intel Core2 pracującym z częstotliwością 3.16GHz. Testy przeprowadzono w arytmetyce podwójnej precyzji.

Zaproponowane w tej pracy kwadratury adaptacyjne będziemy w skrócie nazywać G18, jeśli kwadraturą podstawową kwadratura Gaussa-Legendre'a, oraz L18, gdy kwadraturą podstawową jest kwadratura Lobatto. Kwadraturę z pracy [3], dostępną na stronie internetowej [10], nazwiemy RMS, a kwadraturę z biblioteki [9] – ALGLIB.

Przypomnijmy, że w wypadku algorytmów G18 i L18 można zadać dopuszczalną tolerancję δ błędu bezwzględnego obliczonego przybliżenia. Algorytmy te podają również oszacowanie tego błędu. Jeśli $\bar{\varepsilon}$ jest obliczonym oszacowaniem błędu bezwzględnego, a γ jego rzeczywistą wartością, to idealna sytuacja ma miejsce, gdy $\gamma \le \bar{\varepsilon} \le \delta$. Ponieważ jednak nie zawsze w danej arytmetyce żądaną dokładność da się uzyskać, uznamy, że algorytm działa poprawnie, jeśli $\gamma \le \bar{\varepsilon}$. Algorytmy RMS i ALGLIB liczą całkę najdokładniej jak potrafią i nie oferują oszacowań błędu obliczonego przybliżenia.

W Tabeli 1 porównujemy dokładność (błąd bezwzględny) i czas działania algorytmów G18, L18, RMS i ALGLIB dla testowego zestawu funkcji podcałkowych $f_1, f_2, \ldots, f_9$. Dla algorytmów G18 i L18 podajemy dodatkowo obliczone przez nie oszacowanie błędu. W Tabeli 2 porównujemy kwadraturę adaptacyjną G18 z kwadraturą `quadgk` systemu obliczeń matematycznych *MatLab*. Podane w tej tabeli czasy należy traktować orientacyjnie, gdyż trudno obiektywnie porównać czas działania fragmentu programu w języku C z czasem działania procedury wbudowanej w pewien system obliczeń matematycznych.

Nasze algorytmy adaptacyjne w łatwy sposób mogą podawać, w ilu łącznie punktach przedziału należało obliczyć wartość funkcji podcałkowej (porównaj kody źródłowe procedur w rozdziale 5). Dla przykładu, dla zadanej wartości tolerancji $\delta = 10^{-14}$, wartość funkcji f_1 wystarczyło obliczyć 54 razy, natomiast wartość funkcji f_5 już 1710342 razy.

Przetestujemy jeszcze możliwość zastosowania naszych algorytmów do obliczania całek na płaszczyźnie. Zauważmy, że

$$\int_a^b \int_c^d f(x,y)\,dx\,dy \;=\; \int_a^b g(y)\,dy\,,$$

gdzie

$$g(y) = \int_c^d f(x,y)\,dx\,.$$

Zastosowaliśmy powyższy schemat i prezentowane w pracy kwadratury adaptacyjne, prosząc o przybliżenia całki

$$\int_{-10}^{10}\int_{-10}^{10} \frac{\cos(x^2+y^2+1)}{x^2+y^2+1}\,dx\,dy$$

z błędem nieprzekraczającym 10^{-12}. Oba prezentowane algorytmy adaptacyjne spisały się dobrze, podając w czasie 67 milisekund wynik z błędem mniejszym niż $3{\cdot}10^{-15}$.

4.1 Podsumowanie i wnioski

Jak wynika z zamieszczonych wyników eksperymentów, obie zaproponowane metody, choć bardzo proste, z powodzeniem konkurują ze znacznie bardziej zaawansowanymi algorytmami. Na podstawie przeprowadzonych testów nie można stwierdzić, aby którakolwiek z porównywanych metod była najlepsza. Jeśli przy obliczaniu wartości funkcji podcałkowej nie występuje dzielenie przez 0, najlepiej spisuje się algorytm L18. Metoda RMS również działa bardzo dobrze, lecz momentami kapryśnie. Dzieje się tak za sprawą wbudowanego w nią mechanizmu ekstrapolacji, który często daje bardzo dobre rezultaty (funkcja f_8), ale czasami całkiem zawodzi

(funkcja f_9). Algorytm G18 spisywał się poprawnie dla wszystkich przykładowych całek i działa bardzo podobnie jak oparty na parze kwadratur Gaussa-Kronroda algorytm `quadgk` z systemu *MatLab*. Choć zdaniem autorów ten ostatni zbyt często podaje trochę niepoprawne oszacowanie błędu.

Jak przewidziano w rozdziale 2, wszystkie metody adaptacyjne wykorzystujące otwartą kwadraturę podstawową mniej lub bardziej zawiodły w wypadku funkcji f_{10}, która ma nieciągłą pierwszą pochodną. W tym wypadku jedynie algorytm L18 podał poprawne wyniki.

5 Kwadratura adaptacyjna – kod źródłowy

Procedura 1 *18. punktowa kwadratura podstawowa Gaussa-Legendre'a. Argumenty:* `f` *– funkcja podcałkowa;* `a, b` *– końce przedziału;* `eps` *– epsilon maszynowy;* `*tol, *fe` *– (uaktualniane) tolerancja błędu kwadratury adaptacyjnej i oszacowanie błędów zaokrągleń;* `*n` *– (uaktualniana) liczba wywołań funkcji podcałkowej. Wartością procedury jest przybliżona wartość całki funkcji f na przedziale* $[a, b]$.

```
double BaseQuad(double f(double), double a, double b, double eps,
                double *tol, double *fe, long *n)
{
  #define N 18
  static double x[N] = { // węzły kwadratury ...
     4.217415789534526634992e-03, 2.208802521430112240940e-02, 5.369876675122213039697e-02,
     9.814752051373844215879e-02, 1.541564784698233960626e-01, 2.201145844630262326961e-01,
     2.941244192685786769820e-01, 3.740568871542472452055e-01, 4.576124934791323493789e-01,
     5.423875065208676506211e-01, 6.259431128457527547945e-01, 7.058755807314213230180e-01,
     7.798854155369737673039e-01, 8.458435215301766039374e-01, 9.018527948626215578412e-01,
     9.463012332487778696030e-01, 9.779119747856988775906e-01, 9.957825842104654733650e-01 };
  static double w[N] = { // współczynniki kwadratury ...
     1.080800676324165515667e-02, 2.485727444748489822667e-02, 3.821286512744452826456e-02,
     5.047102205314358278141e-02, 6.127760335573923009226e-02, 7.032145733532532560237e-02,
     7.734233756313262246271e-02, 8.213824187291636149303e-02, 8.457119148157179592033e-02,
     8.457119148157179592033e-02, 8.213824187291636149303e-02, 7.734233756313262246271e-02,
     7.032145733532532560237e-02, 6.127760335573923009226e-02, 5.047102205314358278141e-02,
     3.821286512744452826456e-02, 2.485727444748489822667e-02, 1.080800676324165515667e-02 };
  double b_a = (double)(b-a);
  double mfx = 0.0;          // średnia wartość eps*|f(x)|
  double s = 0.0;            // obliczana całka
  double fx;                 // f(x)
  for (int i=0; i<N; i++) {
    fx = f((b_a)*x[i]+a);
    mfx += fabs(fx);
    s += fx*w[i];
  }
  *n += N;
  mfx = eps*mfx/(double)N;   // szacowanie wpływu błędów zaokrągleń...
  if ( mfx > *tol ) { *tol = mfx; }
  *fe += b_a*mfx;
  return b_a*s;
}
```

Procedura 2 *Rekurencyjna kwadratura adaptacyjna. Argumenty:* `f` *– funkcja podcałkowa;* `a, b` *– końce przedziału;* `w0` *– poprzednia wartość całki;* `tol` *– tolerancja błędu kwadratury adaptacyjnej;* `lev` *– poziom zagłębienia rekurencyjnego;* `eps` *– epsilon maszynowy;* `*err` *– oszacowany błąd bezwzględny obliczonego przybliżenia;* `*n` *– liczba wywołań funkcji podcałkowej. Wartością procedury jest przybliżona wartość całki funkcji f na przedziale* $[a, b]$.

```
double adaptiveR(double f(double), double a, double b, double w0, double tol,
                 int lev, double eps, double *err, long *n)
{
  #define TolM 0.8125        // modyfikator tolerancji błędu
  #define ErM1 0.0375        // parametr szacowania błędu (0.0 dla kw. Lobatto)
  #define ErM2 1.0000        // parametr szacowania błędu (0.0 dla kw. Lobatto)
  #define MaxN 20000000      // maks. liczba wywołań funkcji podcałkowej
  #define MivL 250.0         // MivL*eps = min. długość podprzedziału
  #define MaxL 40            // maks. liczba zagłębień rekurencyjnych
  #define MinL 1             // min. liczba zagłębień rekurencyjnych
  double e;                  // oszacowanie błędu przybliżenia
  double fe = 0.0;           // błąd zaokrągleń
  double m = (a+b)*0.5;
  double w1 = BaseQuad(f,a,m,eps,&tol,&fe,n);
  double w2 = BaseQuad(f,m,b,eps,&tol,&fe,n);
  e = fabs(w1+w2-w0);
  if ( lev >= MinL && ( e < tol || (b-a)<MivL*eps || lev >= MaxL || *n > MaxN ) ) {
    *err += e*(1.0+ErM1*(double)lev) + ErM2*eps*fabs((w2-w1)/(m-a)) + fe;
    return w1+w2;
  }
  else {
```

```
    tol = TolM*tol;
    return adaptiveR(f,a,m,w1,tol,lev+1,eps,err,n) +
           adaptiveR(f,m,b,w2,tol,lev+1,eps,err,n);
  }
}
```

Procedura 3 *Obliczanie wartości epsilon maszynowego (precyzji danej arytmetyki).*

```
double eps_mach()
{
  double x = 0.125;
  double y = 1.0 + x;
  while ( y != 1.0 ) {
    x /= 2.0;
    y = 1.0 + x;
  }
  return x*2.0;
}
```

Procedura 4 *Główna procedura całkująca. To jej należy używać w programach korzystających z proponowanej metody. Argumenty:* `f` *– funkcja podcałkowa;* `a,b` *– końce przedziału;* `tol` *– tolerancja błędu kwadratury adaptacyjnej;* `*err` *– oszacowany błąd bezwzględny obliczonego przybliżenia;* `*n` *– liczba wywołań funkcji podcałkowej. Wartością procedury jest przybliżona wartość całki funkcji f na przedziale $[a,b]$.*

```
double aGL(double f(double), double a, double b, double tol,
           double *err, long *n)
{
  double fe = 0.0;  // zmienna nieużywana; potrzebna do wywołania procedury BaseQuad
  *err = eps_mach();
  *n = 0;
  return adaptiveR(f,a,b,BaseQuad(f,a,b,*err,&tol,&fe,n),tol,1,*err,err,n);
}
```

6 Tabele

		G18		L18		RMS		ALGLIB	
f	tol.	błąd (oszac.)	czas	błąd (oszac.)	czas	błąd	czas	błąd	czas
f_1	10^{-10}	$1{\cdot}10^{-16}$ ($5{\cdot}10^{-16}$)	18	$1{\cdot}10^{-16}$ ($4{\cdot}10^{-16}$)	18				
	10^{-14}	$1{\cdot}10^{-16}$ ($5{\cdot}10^{-16}$)	18	$1{\cdot}10^{-16}$ ($4{\cdot}10^{-16}$)	18	$1{\cdot}10^{-16}$	24	$1{\cdot}10^{-16}$	35
f_2	10^{-10}	$2{\cdot}10^{-16}$ ($6{\cdot}10^{-14}$)	117	$2{\cdot}10^{-16}$ ($7{\cdot}10^{-13}$)	116				
	10^{-14}	$2{\cdot}10^{-16}$ ($8{\cdot}10^{-15}$)	171	$2{\cdot}10^{-16}$ ($2{\cdot}10^{-15}$)	170	$2{\cdot}10^{-16}$	929	$7{\cdot}10^{-16}$	257
f_3	10^{-10}	$9{\cdot}10^{-11}$ ($1{\cdot}10^{-10}$)	17	$9{\cdot}10^{-11}$ ($1{\cdot}10^{-10}$)	17				
	10^{-14}	$9{\cdot}10^{-11}$ ($1{\cdot}10^{-10}$)	17	$9{\cdot}10^{-11}$ ($1{\cdot}10^{-10}$)	17	$8{\cdot}10^{-11}$	286	$8{\cdot}10^{-11}$	165
f_4	10^{-10}	$3{\cdot}10^{-16}$ ($1{\cdot}10^{-10}$)	657	$1{\cdot}10^{-16}$ ($1{\cdot}10^{-15}$)	696				
	10^{-14}	$3{\cdot}10^{-16}$ ($8{\cdot}10^{-15}$)	696	$1{\cdot}10^{-16}$ ($1{\cdot}10^{-15}$)	696	$6{\cdot}10^{-16}$	788	$1{\cdot}10^{-16}$	1430
f_5	10^{-4}	$2{\cdot}10^{-6}$ ($3{\cdot}10^{-5}$)	72	– (–)	–				
	10^{-6}	$6{\cdot}10^{-8}$ ($3{\cdot}10^{-7}$)	180	– (–)	–				
	10^{-8}	$2{\cdot}10^{-9}$ ($8{\cdot}10^{-9}$)	597	– (–)	–	$7{\cdot}10^{-9}$	806		
	10^{-10}	$2{\cdot}10^{-11}$ ($2{\cdot}10^{-10}$)	3950	– (–)	–				
	10^{-12}	$2{\cdot}10^{-13}$ ($4{\cdot}10^{-12}$)	25600	– (–)	–			$5{\cdot}10^{-13}$	149000
	10^{-14}	$3{\cdot}10^{-15}$ ($8{\cdot}10^{-14}$)	162000	– (–)	–				
	0	$2{\cdot}10^{-16}$ ($8{\cdot}10^{-15}$)	754000	– (–)	–				
f_6	10^{-5}	$2{\cdot}10^{-6}$ ($8{\cdot}10^{-6}$)	589	– (–)	–	$7{\cdot}10^{-6}$	810		
	10^{-10}	$7{\cdot}10^{-11}$ ($6{\cdot}10^{-10}$)	67200	– (–)	–			$5{\cdot}10^{-10}$	145000
	10^{-14}	$1{\cdot}10^{-13}$ ($8{\cdot}10^{-12}$)	75800	– (–)	–				
f_7	10^{-10}	$7{\cdot}10^{-14}$ ($4{\cdot}10^{-11}$)	256	$2{\cdot}10^{-12}$ ($1{\cdot}10^{-11}$)	286	$6{\cdot}10^{-12}$	742	$2{\cdot}10^{-11}$	241
	10^{-14}	$4{\cdot}10^{-12}$ ($2{\cdot}10^{-11}$)	319	$4{\cdot}10^{-12}$ ($5{\cdot}10^{-12}$)	347				
f_8	10^{-10}	$8{\cdot}10^{-8}$ ($8{\cdot}10^{-8}$)	818	– (–)	–			–	–
	10^{-14}	$8{\cdot}10^{-8}$ ($8{\cdot}10^{-8}$)	882	– (–)	–	$9{\cdot}10^{-13}$	180		
f_9	10^{-10}	$5{\cdot}10^{-12}$ ($8{\cdot}10^{-11}$)	13700	$1{\cdot}10^{-12}$ ($1{\cdot}10^{-10}$)	13700	$4{\cdot}10^{-3}$	1160		
	10^{-14}	$5{\cdot}10^{-12}$ ($3{\cdot}10^{-11}$)	14600	$5{\cdot}10^{-13}$ ($9{\cdot}10^{-12}$)	14700			$5{\cdot}10^{-13}$	15700
f_{10}	10^{-10}	$4{\cdot}10^{-7}$ ($6{\cdot}10^{-11}$)	2630	$3{\cdot}10^{-12}$ ($3{\cdot}10^{-11}$)	3300	$1{\cdot}10^{-4}$	540	$4{\cdot}10^{-7}$	3490
	10^{-14}	$4{\cdot}10^{-7}$ ($1{\cdot}10^{-14}$)	4110	$8{\cdot}10^{-16}$ ($2{\cdot}10^{-15}$)	5170				

Tabela 1: Porównanie dokładności i efektywności (czas w mikrosekundach) adaptacyjnych algorytmów G18, L18, RMS i ALGLIB na przykładach całek funkcji f na odcinku $[-1,1]$. Znak „–" oznacza, że algorytm nie potrafił obliczyć danej całki.

The comparison of accuracy and efficiency (time in microseconds) of the adaptive algorithms G18, L18, RMS and ALGLIB in the case of the integrals of the function f over the interval $[-1,1]$. The "–" sign means that the algorithm failed to compute the given integral.

		G18		MATLAB	
f	tol.	błąd (oszac.)	czas	błąd (oszac.)	czas
f_1	10^{-10}	$1{\cdot}10^{-16}$ ($5{\cdot}10^{-16}$)	18	$1{\cdot}10^{-16}$ ($3{\cdot}10^{-17}$)	568
	10^{-14}	$1{\cdot}10^{-16}$ ($5{\cdot}10^{-16}$)	18	$1{\cdot}10^{-16}$ ($3{\cdot}10^{-17}$)	568
f_3	10^{-10}	$9{\cdot}10^{-11}$ ($1{\cdot}10^{-10}$)	17	$1{\cdot}10^{-10}$ ($8{\cdot}10^{-11}$)	1590
	10^{-14}	$9{\cdot}10^{-11}$ ($1{\cdot}10^{-10}$)	17	$8{\cdot}10^{-11}$ ($2{\cdot}10^{-11}$)	16900
f_5	10^{-4}	$2{\cdot}10^{-6}$ ($3{\cdot}10^{-5}$)	72	$2{\cdot}10^{-4}$ ($8{\cdot}10^{-5}$)	863
	10^{-6}	$6{\cdot}10^{-8}$ ($3{\cdot}10^{-7}$)	180	$5{\cdot}10^{-8}$ ($4{\cdot}10^{-7}$)	1650
	10^{-8}	$2{\cdot}10^{-9}$ ($8{\cdot}10^{-9}$)	597	$1{\cdot}10^{-10}$ ($9{\cdot}10^{-9}$)	3750
	10^{-10}	$2{\cdot}10^{-11}$ ($2{\cdot}10^{-10}$)	3950	$9{\cdot}10^{-13}$ ($9{\cdot}10^{-11}$)	16300
	10^{-12}	$2{\cdot}10^{-13}$ ($4{\cdot}10^{-12}$)	25600	$4{\cdot}10^{-15}$ ($9{\cdot}10^{-13}$)	280000
	10^{-14}	$3{\cdot}10^{-15}$ ($8{\cdot}10^{-14}$)	162000	$2{\cdot}10^{-15}$ ($4{\cdot}10^{-13}$)	716000
	0	$2{\cdot}10^{-16}$ ($8{\cdot}10^{-15}$)	754000	$2{\cdot}10^{-15}$ ($4{\cdot}10^{-13}$)	716000
f_8	10^{-10}	$8{\cdot}10^{-8}$ ($8{\cdot}10^{-8}$)	818	$1{\cdot}10^{-12}$ ($5{\cdot}10^{-11}$)	1630
	10^{-14}	$8{\cdot}10^{-8}$ ($8{\cdot}10^{-8}$)	882	$1{\cdot}10^{-8}$ ($1{\cdot}10^{-8}$)	13500
f_{10}	10^{-10}	$4{\cdot}10^{-7}$ ($6{\cdot}10^{-11}$)	2630	$4{\cdot}10^{-7}$ ($6{\cdot}10^{-11}$)	3450
	10^{-14}	$4{\cdot}10^{-7}$ ($1{\cdot}10^{-14}$)	4110	$4{\cdot}10^{-7}$ ($2{\cdot}10^{-14}$)	4607

Tabela 2: Porównanie dokładności i efektywności (czas w mikrosekundach) algorytmów G18 i `quadgk` z systemu *MatLab* na przykładach całek funkcji f na odcinku $[-1, 1]$.

The comparison of accuracy and efficiency (time in microseconds) of the algorithm G18 and the *MatLab* `quadgk` algorithm in the case of the integrals of the function f over the interval $[-1, 1]$.

Literatura (References)

[1] G. Dahlquist i Å. Björck, *Numerical Methods in Scientific Computing, Volume 1*, Society for Industrial Mathematics, 2008.

[2] P. J. Davis i P. Rabinowitz, *Methods of Numerical Integration (2nd edition)*, Academic Press, New York, 1984.

[3] P. Favati, G. Lotti i F. Romani, *Algorithm 691: Improving QUADPACK automatic integration routines*, ACM Trans. Math. Soft **17** (1991), 218–232.

[4] W. Gander i W. Gautschi, *Adaptive Quadrature – Revisited*, BIT **40** (2000), 84–101.

[5] D. Kincaid i W. Cheney, *Analiza numeryczna*, WNT 2006.

[6] L. F. Shampine, *Vectorized Adaptive Quadrature in Matlab*, J. Comput. Appl. Math. **211** (2008), 131–140.

[7] L. F. Shampine, R. C. Allen, Jr. i S. Pruess, *Fundamentals of Numerical Computing*, Wiley, New York, 1997.

[8] `http://alpha.uwb.edu.pl/amicke/globalq.shtml`

[9] `http://www.alglib.net/`

[10] `http://www.codecogs.com/code/maths/calculus/quadrature/adaptive.php`

[11] `http://www.nag.com/`

Steganokryptografia typu „Grayscale Image”

Grayscale Image Steganocryptography

Czesław Kościelny

Wrocławska Wyższa Szkoła Informatyki Stosowanej
ul. Wejherowska 28, 54-239 Wrocław

Treść. Przedstawiono nową metodę steganokryptografii, polegającą na przekształceniu pliku dowolnego typu na plik graficzny „Grayscale Image” o formacie TIF lub BMP. Wiadomość, zawarta w przekształcanym pliku jest najpierw szyfrowana w taki sposób, że plik po zaszyfrowaniu zawiera wyłącznie znaki kodu ASCII o numerach od 0 do 31. Następnie tak utworzony kryptogram jest zamieniany na plik graficzny, zwany steganokryptogramem, którego treścią jest obraz szachownicy utrzymany w skali szarości. Efektywność metody polega na tym, że steganokryptogram ma tego samego rzędu rozmiar, co plik, którego treść jest ukrywana.

S**łowa kluczowe**: Steganografia, kryptografia, steganokryptografia, pliki graficzne w skali szarości

Abstract. In the paper, a steganocryptographic method converting an arbitrary format disk file into grayscale image of the TIF or BMP format has been presented. A message contained in the file to be converted is encrypted into cryptogram first, in which characters with numbers from 0 to 31are contained. Then, the obtained kryptogram is transformed into a grayscale image file, presenting a chessboard, is transformed. This is so-called steganocryptogram. The method is effective, which means that the size of a message, contained in the steganocryptogram is comparable with the size of the latter.

Keywords: Grayscale image, steganography, cryptography, steganocryptography

1. Wstęp

Znana od starożytności metoda ukrywania tajnych wiadomości, zwana steganografią, poczynając od roku 1995 zaczęła być szeroko stosowana w informatyce jako alternatywa dla kryptografii oraz jako narzędzie do zabezpieczania plików przed kopiowaniem. W zastosowaniu do plików komputerowych steganografia polegała na ukrywaniu pliku z tajną wiadomością w nieużywanych bitach pliku, zawierającym wiadomość nieistotną. Dlatego też tajna wiadomość miała rozmiar od kilku do kilkunastu procent rozmiaru pliku, w którym sekretną informację ukrywano. Stosunkowo niedawno pokazano [1, 2] w jaki sposób efektywnie ukrywać wiadomości, tzn. tak, aby rozmiar pliku z ukrytą wiadomością był porównywalny z rozmiarem pliku zawierającego wiadomość tajną. Wprowadzono też pojęcie steganokryptografii, polegającej na tworzeniu steganogramów szyfrowanych, czyli steganokryptogramów. W pracy opisano oryginalną metodę generowania steganokryptogramów i odzyskiwania zawartych w nich wiadomości.

2. System kryptograficzny

Aby wygenerować steganokryptogram należy najpierw zaszyfrować plik z tajną wiadomością. Zastosowano tu jeden wariantów metody [3, 4], stosującej nieliniowe przekształcenie w postaci zapisu liczb w systemach liczbowych o różnych podstawach. System kryptograficzny składa się z trzech algorytmów: generowanie tajnego klucza, szyfrowanie pliku i deszyfrowanie pliku. Oryginalność metody polega nie tylko na zastosowanym przekształceniu kryptograficznym, ale też na wykonywaniu operacji szyfrowania i deszyfrowania na całym pliku, bez potrzeby dzielenia pliku na bloki.

Algorytm generowania klucza:
Wejście: liczba *kl*, oznaczająca liczbę bitów, czyli długość klucza.
Krok 1. Wygenerować *kl* -bitową liczbę systemu dziesiątkowego z niezerowym bitem o wadze $2^{(kl-1)}$

Algorytm szyfrowania:
Wejście: nazwa pliku *fn,* klucz szyfrujący *key.*
Krok 1. Przeczytać bajty pliku *fn* do listy *fp*, otrzymując $fp=[b_1,b_2,...,b_{fs}]$
Krok 2. Wyznaczyć liczbę
$N = b_1+b_2256+b_3256^2+,...,+b_{(fs-1)}256^{(fs-2)}+bfs256^{(fs-1)}+256^{fs}key$
Krok 3. Dokonać konwersji liczby N na liczbę systemu liczbowego o podstawie 32, czyli obliczyć *fc* oraz współczynniki $c_1,i=1,2,...,fc$ według wzoru
$N=c1+c232+c3322+,...,+cfc32(fc-1).$
Krok 4. Wpisać listę bajtów $[c_1,c_2,...cf_c]$ do pliku *fn.*

Algorytm deszyfrowania:
Wejście: nazwa pliku *fn*, klucz szyfrujący *key.*
Krok 1. Przeczytać bajty pliku *fn* do listy *fc*, otrzymując $fc=[c_1,c_2,...cf_c]$
Krok 2. Wyznaczyć liczbę
$N=c_1+c_232+c_332^2+,...,+c_{fc}32^{(fc-1)}+key$

Krok 3. Dokonać konwersji liczby *N* na liczbę systemu liczbowego o podstawie 256, czyli obliczyć *fs* oraz współczynniki *bi,i*=1,2,...,*fs* według wzoru
$N=b_1+b_2256+b_3256^2+,...,+b_3^{(fs-1)}256^{(fs-2)}+b_{fs}256^{(fs-1)}+256^{fs}$
Krok 4. Wpisać listę bajtów [*b1,b2,...,bfs*] do pliku *fn*.

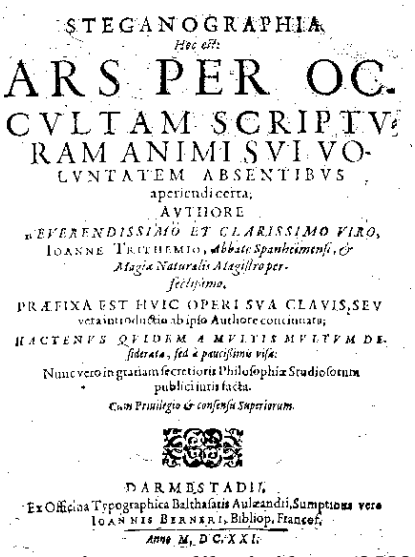
Prosty przykład:
Zakładając, że treścią pliku tekstowego jest napis TRITHEMIUS, a klucz jest w postaci 32-bitowej liczby = 2882724882, to algorytm szyfrowania będzie miał przebieg:
Krok 1. fp = [84, 82, 73, 84, 72, 69, 77, 73, 85, 83],
Krok 2. N = 1602455492893187086118466,
Krok 3. lista współczynników [2, 18, 27, 12, 7, 20, 30, 8, 5, 10, 19, 18, 20, 10, 13, 10, 1],
Krok 4. lista z poprzedniego kroku zostanie wpisana do pliku kryptogramu. Oznacza to, że steganokryptogramem 10-znakowego napisu TRITHEMIUS będzie 17 znaków
niedrukowalnych o numerach, pokazanych w liście.

Podczas deszyfrowania otrzyma się:
Krok 1. fc = [2, 18, 27, 12, 7, 20, 30, 8, 5, 10, 19, 18, 20, 10, 13, 10, 1],
Krok2. N = 1602455492893189968843348,
Krok 3. Otrzymana w tym kroku lista bajtów, [84, 82, 73, 84, 72, 69, 77, 73, 85, 83, 1] ,
bez ostatniego bajtu jest taka sama jak w 1. kroku procedury szyfrowania.
Krok 4. Lista z kroku 3., bez ostatniego bajtu, zostaje wpisana do pliku wyjściowego.

3. Przekształcanie kryptogramu w steganokryptogram

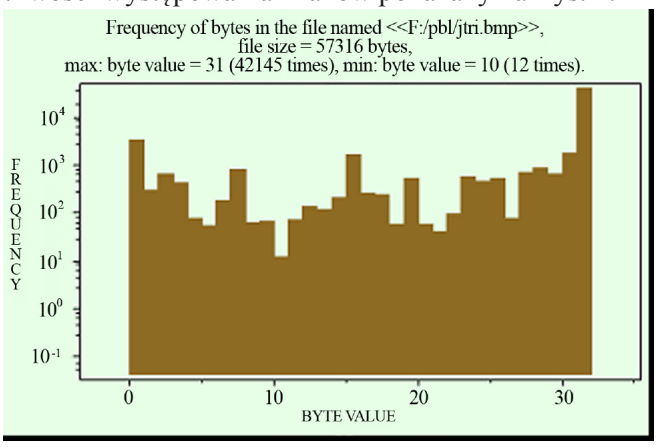
Ten etap przetwarzania plików polega na „narysowaniu" szachownicy przy pomocy bitów, zawartych w kryptogramie pliku, stanowiącym wiadomość tajną. Wymaga to dobrej znajomości pakietów bibliotecznych używanego środowiska programistycznego, dotyczących przetwarzania obrazów. Dzięki metodzie szyfrowania, generującej kryptogramy składające się wyłącznie ze znaków o numerach od 0 do 31, steganokryptogramy plików o różnych formatach są do siebie bardzo podobne i wyraziste.

Opis przykładu generowania steganokryptogramu i odzyskiwania ukrytej wiadomości
W tym przykładzie wiadomością do ukrycia jest strona tytułowa XV -wiecznego dzieła o steganografii i ta wiadomość jest zapisana w pliku jtri.bmp o objętości 35822 bajty (Rys. 1).


STEGANOGRAPHIA
Hoc eſt:
ARS PER OC.
CVLTAM SCRIPTV
RAM ANIMI SVI VO-
LVNTATEM ABSENTIBVS
aperiendi certa;
AVTHORE
R. EVERENDISSIMO ET CLARISSIMO VIRO,
IOANNE TRITHEMIO, Abbate Spanheimenſi, &
Magiæ Naturalis Magiſtro per-
ſpicuiſſimo.
PRÆFIXA EST HVIC OPERI SVA CLAVIS, SEV
vera introductio ab ipſo Authore concinnata;
HACTENVS QVIDEM A MVLTIS MVLTVM DE-
ſiderata, ſed a pauciſſimis viſa:
Nunc vero in gratiam ſecretioris Philoſophiæ Studioſorum
publici iuris facta.
Cum Priuilegio & conſenſu Superiorum.
DARMESTADII,
Ex Officina Typographica Balthaſaris Aulæandri, Sumptibus vero
IOANNIS BERNERI, Bibliop. Francof.
Anno M. DC. XXI.

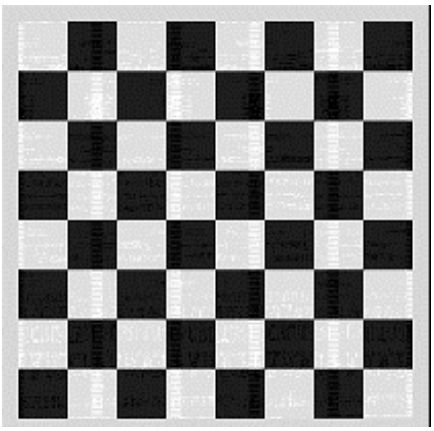
Rys. 1. Obraz zapisany w pliku jtri.bmp (35822 bajty)
Fig. 1. The image contained in the file jtri.bmp (35822 bytes)

Jak w każdym pliku graficznym, także i w tym pliku występują prawie wszystkie znaki kodu ASCII. Podstawową charakterystykę statyczną pliku z Rys.1 pokazano na Rys.2. W procedurze szyfrowania zastosowano 1025 bitowy klucz
key = 258336886841438853633549038007191828532868056745471672735362998052075538656810006153745439090605990420768146034351950576375613004215612669834810959945854795015629329485604310236232183115836846059598901589193318530700923353688489418223071772791792353624045605496946700965633275730897048879057257788360000545045,
i w efekcie otrzymano kryptogram pliku z Rys. 1. o objętości 57 316 bajtów, posiadający histogram częstotliwości występowania znaków pokazany na Rys. 2.



Rys. 2. Częstotliwość występowania znaków kodu ASCII w zaszyfrowanym pliku z Rys. 1.
Fig. 2. The frequency of bytes of the encrypted file from Fig. 1.

Ostatecznie wygenerowany steganokryptogram jest plikiem graficznym TIF, ma rozmiar 16 384 bajty i zawiera obraz, pokazany na Rys. 3.



Rys. 3. Obraz steganokryptogramu pliku z Rys. 1.
Fig. 3. The image of the steganocryptogram of the file shown in Fig. 1.

Podobne eksperymenty przeprowadzono z plikami o formatach TXT, JPG, BPP, TIFF, GIF, MID, WAV, RTF, AVI, uzyskując steganokryptogramy bardzo podobne do obrazu, pokazanego na Rys. 3.
Przykład zrealizowano za pomocą dość skomplikowanego programu, którego opis będzie przedmiotem oddzielnej publikacji.

4. Podsumowanie i wnioski

Przedstawiono prosty matematycznie sposób wyjątkowo skutecznego zabezpieczania plików dyskowych przed nieupoważnionym dostępem. Chociaż algorytmy kryptograficzne są bardzo proste, to wymagają wykonywania operacji na ogromnych liczbach całkowitych. Jeśli np. generuje się steganokryptogram pliku 100 kilobajtowego, to w procesie szyfrowania zawartość tego pliku jest zamieniana na liczbę liczącą ponad 240 824 cyfry. Dlatego też, chociaż czasy realizacji procedur kryptograficznych nie zależą od klucza, który może mieć praktycznie dowolną długość, to dla dużych plików trwają długo. Zaproponowana metoda nadaje się więc raczej do plików o rozmiarach nie przekraczających stu kilobajtów. Jeśli zaś chodzi o obraz steganokryptogramu, to „narysowanie" takiego obrazu zależy wyłącznie od umiejętności programisty i tutaj autor nie wykazał się nadmiarem pomysłowości. W każdym razie steganokryptogramy nie są podobne do typowych kryptogramów, mogą przypominać obiekty deterministyczne, a ponieważ są regularnymi plikami graficznymi, to bez znajomości niniejszej pracy nie ma możliwości wykrycia zawartych w nich tajnych wiadomości.

Według wiedzy autora, w dostępnej literaturze nie można znaleźć publikacji innych autorów, dotyczących steganokryptografii i algorytmów kryptograficznych z przekształceniem zapisu wiadomości do zaszyfrowania za pomocą konwersji i systemów liczbowych.

Literatura (References)

[1] Cz. Kościelny, *Steganocryptography with Maple* 8. 2003. http://www.maplesoft.com/applications/view.aspx?SID=4348
[2] Cz. Kościelny, *The MLA Steganography. 2006.* http://www.maplesoft.com/applications/view.aspx?SID=1707
[3] Cz. Kościelny, *A Symmetric-Key Block Cipher Generating Cryptograms Containing Characters Belonging to the Definite Set*. 2008. http://www.maplesoft.com/applications/view.aspx?SID=5646
[4] Cz. Kościelny, *Grayscale Image Steganography.* 2008. http://www.maplesoft.com/applications/view.aspx?SID=6878

Sieci neuronowe jako narzędzie do predykacji zachowań giełdy papierów wartościowych

Neural networks as a tool to predict the behaviours of exchange stock markets

Karolina Plawgo, Marian Czerwiński

Wrocławska Wyższa Szkoła Informatyki Stosowanej
ul.Wejherowska 28, 54-239 Wrocław

Treść. Celem badań było znalezienie odpowiedzi na pytanie czy rynek akcji zachowuje się w przypadkowy sposób, nie posiadając właściwie przewidywalnych trendów, czy też trendy te mogą być prognozowane. W celu uzyskania odpowiedzi na to pytanie zostały skonstruowane odpowiednie sieci neuronowe w oparciu o pakiet programu Qnet 2000 [1], które następnie poddano trenowaniu i testowaniu. Z przeprowadzonych analiz wynika, iż relatywnie proste modele mogą dawać dobre wyniki w prognozowaniu indeksu wszystkich spółek WIG-u. Interesującą informacją jest brak możliwości predykcji dla indeksu 20-stu spółek (WIG20) oraz wniosek, że nie można oczekiwać odpowiedzi poprawnej od sieci co do zachowania się cen dla wybranych spółek na podstawie zachowania się cen pozostałych.

Słowa kluczowe: sieci neuronowe, sztuczna inteligencja, prognozowanie wskaźników finansowych, Warszawska Giełda Papierów Wartościowych.

Abstract. The purpose of the researches was to find the reply to a question whether the stock market behaves in an incidental way, without any predictable trends or whether these trends might be predictable. In order to find the reply for these questions, the proper neural networks were build based on Qnet 2000 programme [1], and they were subsequently trained and tested. The analysis which were carried out revealed that the relatively simple models might give good results in forecasting Warsaw Stock Exchange Market Index (WIG). Interesting information is the fact, that there is no possibility to predict the twenty biggest companies' index (WIG20) and the conclusion that we cannot expect from the neural networks the right answer concerning changes of a company's share price based on changes in share prices of the rest of the companies.

Keywords: neural networks, artificial intelligence, forecasting financial index, Warsaw Exchange Stock Market

1. Wprowadzenie

Obserwuje się coraz większą różnorodność metod, które są stosowane do analizy danych finansowych, a w szczególności do analizy finansowych szeregów czasowych. Zaczyna się sięgać po bardziej skomplikowane (niż analiza techniczna) metody prognozowania finansowego. Z reguły wymagają one stosowania szybkich komputerów i specjalistycznego oprogramowania. Jedną z przyczyn tego faktu jest niewątpliwie to, że rozwój technologii komputerowej umożliwia implementację nawet bardzo skomplikowanych metod matematycznych.

Sztuczne sieci neuronowe (dalej SN) stanowią jedną z najbardziej dynamicznie rozwijających się obecnie gałęzi sztucznej inteligencji. Obserwowane jest, w prowadzonych badaniach naukowych, przesuwanie akcentu z badań podstawowych w kierunku badań związanymi z konkretnymi zastosowaniami. Sieci neuronowe okazały się wygodnym narzędziem, przydatnym do realizacji bardzo wielu różnych praktycznych zadań. W istocie są one z powodzeniem stosowane w niezwykle szerokim zakresie problemów, w tak różniących się od siebie dziedzinach jak finanse, medycyna, zastosowania inżynierskie, geologia czy fizyka. Szeroki zakres zastosowań sieci neuronowych obejmuje również zagadnienia z zakresu nauk ekonomicznych, które do tej pory badano głównie za pomocą modeli statystycznych, ekonometrycznych, czy optymalizacyjnych. Należy mieć na uwadze fakt, iż krytycznymi parametrami przy rozwiązywaniu skomplikowanych problemów optymalizacyjnych są często ograniczenia czasowe bądź ograniczenia sprzętowe. Sieci neuronowe stosowane do rozwiązywania tego typu problemów osiągają bardzo dobre rezultaty w krótkim czasie i przy ograniczonych wymaganiach sprzętowych. Dodatkowo, bardzo istotną zaletą sieci neuronowych jest uniwersalność stosowanej metodologii rozumiana jako niezależność od szczegółowej definicji oraz danych problemu (w ramach określonej grupy problemów). Sieci neuronowe posiadają również zdolność uogólniania zdobytej wiedzy oraz ewolucyjnej auto-poprawy efektywności swojego działania. Konsekwencją takiego stanu rzeczy jest możność rozwiązywania przez sieć neuronową po okresie nauki nie tylko problemów treningowych ale również – co istotne – nieznanych problemów pokrewnych do tych na których była trenowana [3].

Efektywność sieci neuronowych wynika ze stosowania masowego przetwarzania równoległego w oparciu o bardzo dużą liczbę nieskomplikowanych elementów przetwarzających nazwanych neuronami.

Zastosowanie wielu prostych elementów przetwarzających funkcjonujących w trybie równoległym w miejsce jednego czy kilku wyspecjalizowanych procesorów powoduje, że obok zwiększania szybkości przetwarzania uzyskuje się efekt odporności na ewentualne błędy czy zakłócenia w funkcjonowaniu poszczególnych neuronów. Odporność na błędy wynika z zamierzonej redundancji elementów przetwarzających, z implementacji mechanizmów kontrolnych i autokorekcyjnych w sieciach neuronowych [7].

Uniwersalność metodologii, zdolność generalizacji i autokorekcji, relatywnie małe wymagania czasowe i sprzętowe czynią z sieci neuronowych atrakcyjne i efektywne narzędzie do rozwiązywania wielu problemów np.: przewidywania zachowań (predykcji), grupowania danych, rozpoznawania obrazów, optymalizacji i innych. Metody analizy finansowych szeregów czasowych przy użyciu sieci neuronowych należy zaliczyć do grupy metod ilościowych wywodzących się z szeroko rozumianego pojęcia dochodu. Siłą napędową, która spowodowała ich rozwój, była chęć stworzenia metody prognozowania wskaźników finansowych (w szczególności kursów akcji), których stosowanie na rynku przynosiłoby ponadprzeciętne dochody [5].

Wspólną cechą metod wykorzystywanych do prognozowania finansowego jest założenie braku efektywności rynku, nawet w słabej formie, tzn. zakłada się, że informacje o przeszłych cenach finansowych (np. kursach akcji) nie są odzwierciedlone w cenie, czyli ma sens określenie prognozy ceny, której wykorzystanie prowadziłoby do uzyskania ponadprzeciętnych dochodów [2].

W tej pracy opisane zostaną wyniki zastosowania SN do badania predykcji indeksów giełdowych na przykładzie Warszawskiej Giełdy Papierów wartościowych (WGPW). Efektywne narzędzia do przewidywania indeksów giełdy a w konsekwencji kursu akcji są z pewnością informacją wielce pożądaną przez inwestorów. Indeks giełdowy jest to podstawowa charakterystyka giełdy. Wartość indeksu giełdowego jest najprostszą odpowiedzią na podstawowe pytanie inwestora: „co się dzieje na giełdzie?".

Wykorzystanie odpowiednio zbudowanych i wyuczonych sieci neuronowych może przyczynić się do usprawnienia kontrolowania i sterowania złożonymi procesami występującymi także w innych rodzajach działalności gospodarczej.

2. Konstrukcja sieci neuronowej do problemu predykcji

Rozpoczynając budowę sieci musimy określić kilka istotnych jej parametrów. Zasadniczą kwestią jest dobranie odpowiedniej ilości neuronów. Zbyt mała ilość neuronów doprowadzi do braku zbieżności sieci. W zasadzie dla większości problemów nie ma jednoznacznych metod wyznaczenia minimalnej ilości neuronów. Ilość neuronów wejściowych i wyjściowych jest determinowana przez rozwiązywany problem. Ustala się ją doświadczalnie [6].

Sieci neuronowe mają budowę warstwową, która daje się łatwo zdefiniować i szczegółowo opisać nawet w tym przypadku, gdy ilość neuronów wchodzących w jej skład jest bardzo duża. Do pełnego zdefiniowania sieci warstwowej wystarczy podać ilość neuronów w każdej warstwie. Przy innej, w szczególności tzw. dowolnej architekturze sieci - wysiłek związany z definicją tej topologii może być nieakceptowalnie duży. Określenie liczby warstw ukrytych i ilości neuronów znajdujących się w tych warstwach nie jest prostym zadaniem. Jako punkt wyjścia można przyjąć sieć z jedną warstwą ukrytą, zawierającą taką ilość neuronów, która jest równa połowie sumy ilości neuronów wejściowych i ilości neuronów wyjściowych, jednak na ogół najlepsze wyniki otrzymuje się, wybierając te ilości w sposób empiryczny [8].

Kolejnym problemem wymagającym wyjaśnienia jest kwestia struktury połączeń między neuronami. Jak wiadomo, struktura połączeń ma wpływ na działanie sieci, więc jej racjonalny wybór może znacznie przyspieszyć proces jej uczenia.

W celu znalezienia odpowiedniej struktury, można odwołać się do następującego rozumowania. Jednym z ważnych powodów stosowania sieci neuronowych jest fakt, że dla zadań, które chcemy powierzyć do rozwiązania budowanej sieci nie znamy dobrej algorytmicznej metody ich rozwiązania. Jednak skoro nie wiemy, jak postawione zadanie trzeba rozwiązać (chcemy, żeby sieć sama to ustaliła na podstawie przykładów podawanych w trakcie uczenia), to zwykle nie wiemy również z góry, jakie drogi przesyłania sygnałów będą przy tym rozwiązaniu potrzebne, a jakie nie. Nie potrafimy zatem a priori powiedzieć, które połączenia w sieci będą potrzebne, a które nie. Takim rozwiązaniem jest połączenie typu „każdy z każdym" [8]. Przy spełnieniu tej zasady każdy neuron ukryty i każdy neuron wyjściowy jest połączony z każdym neuronem z warstwy poprzedniej. Taka metoda jest kosztowna (czas obliczeń i konieczność posiadania dużego zapasu pamięci).

Dzięki zdefiniowaniu połączeń według zasady „każdy z każdym" niczego z góry nie wyklucza się i dopiero proces uczenia formuje ostateczną strukturę sieci, ustalając niezerowe wartości współczynników wagowych tylko na niektórych (na ogół relatywnie nielicznych) drogach przepływu sygnałów. Te połączenia można potraktować jako nie istniejące i usunąć z sieci.

Musimy rozstrzygnąć także, czy badany problem

jest o charakterze liniowym czy nieliniowym. Stosujemy inne podejście w konstrukcji sieci do problemów liniowych, a inne do nieliniowych. Popełnienie błędu w tym zakresie jest niebezpieczne. Może bowiem okazać się, że badany system jest nieliniowy, jednak okresowo zachowuje się liniowo. Jeżeli analityk zna jedynie okresy liniowe to zastosuje sieć liniową, której dopasowanie będzie doskonałe do rozwiązywanego problemu. Przez pewien okres prognozy będą także bardzo dobre. Po pewnym czasie okaże się jednak, że prognozy pogarszają się – a wkrótce, że nie nadają się do dalszych analiz.

Stosunkowo bezpiecznie jest założyć, że procesy na rynku kapitałowym są nieliniowe. W zdecydowanej większości przypadków jest to prawda. W pozostałych przypadkach sieć nieliniowa i tak dobrze poradzi sobie z liniowymi problemami. Kosztem będzie jedynie dłuższy czas analiz [9].

Pojawia się tutaj problem zbieżności sieci – czyli możliwości nauczenia się problemu przez sieć. Podejście liniowe dla problemów nieliniowych nie doprowadzi do osiągnięcia zbieżności. Sieć nie będzie w stanie się go nauczyć. Sieć nieliniowa nauczy się bez trudu problemu liniowego – jednak najczęściej kosztem dłuższego czasu pracy i (czasami) nieznacznego pogorszenia zdolności prognostycznych. Oczywiście nie jest to jedyny warunek osiągnięcia zbieżności przez sieć.

Budując sieć musimy sprawdzić obszar zmienności analizowanego szeregu. Z niego wynikać będzie postać funkcji aktywacji. Jeżeli analizujemy problem liniowy wybierzemy liniową funkcję aktywacji (purelin). Dla problemów nieliniowych wybierzemy jedną z nieliniowych: sigmoidalną, tangens hiperboliczny lub inną. Sigmoidę wybierzemy, gdy wartości badanego szeregu są wyłącznie dodatnie. Tangens hiperboliczny będzie lepszy, gdy wartości są ujemne i dodatnie. Błędny wybór funkcji aktywacji będzie powodował problemy ze zbieżnością i generalizacją sieci.

Przechodząc do problemu wyboru algorytmu uczenia sieci neuronowej należy odnotować fakt, że najczęściej używanym algorytmem uczenia jest metoda wstecznej propagacji błędu (BP). Po drugie BP jest metoda powolną, ale dość pewną - to znaczy za pomocą tej metody można rozwiązać praktycznie każde zadanie, podczas gdy metody alternatywne w niektórych przypadkach gwarantują bardzo dużą szybkość uczenia jednakże niską praktyczną użyteczność [4].

Podczas eksploatacji sieci neuronowej największe znaczenie ma to, aby sieć osiągała jak najmniejszy błąd związany z prezentacją nowego przypadku. Inaczej mówiąc, najbardziej pożądaną cechą sieci jest zdolność do generalizacji wiedzy na nowe przypadki. Tymczasem w rzeczywistości sieć uczona jest w sposób zapewniający minimalizację błędu wyłącznie dla zbioru uczącego, co nie jest tym samym co minimalizacja rzeczywistego błędu. Związane jest to ze zjawiskiem tzw. przeuczenia sieci (problem nadmiernego dopasowania). Sieć z większa liczbą wag może modelować bardziej złożone funkcje i z tego powodu ma większą skłonność zbytniego dopasowania się do danych. Sieć z mniejszą liczbą wag może z kolei nie być dostatecznie dobrym narzędziem do opisu występującej w rzeczywistości zależności.

Na przykład, sieć nie posiadająca warstw ukrytych może modelować wyłącznie proste zależności liniowe. Natomiast sieć o zbyt dużej liczbie warstw i zbyt dużej ilości neuronów w warstwach ukrytych będzie miała skłonność do uczenia się „na pamięć" całego zbioru uczącego. W związku z tym pojawia się pytanie dotyczące sposobu wyboru sieci o właściwej złożoności.

Rozwiązaniem powyżej zasygnalizowanego problemu może być użycie procesu walidacji. Polega on na tym, że pewna liczba przypadków uczących jest zaliczana do oddzielnej grupy. Dane znajdujące się w tej oddzielnej grupie nie są bezpośrednio stosowane w trakcie uczenia sieci. Natomiast są one wykorzystane do przeprowadzenia niezależnej kontroli postępów algorytmu uczenia [8].

Jeśli jakość odpowiedzi sieci na dane uczące i na dane walidacyjne nie jest przynajmniej w przybliżeniu identyczna, to najprawdopodobniej podział przypadków między dwa zbiory był obciążony jakąś ukrytą tendencją - zaleca się w takim przypadku przerwanie uczenia i ponowny (losowy) podział posiadanych danych na część uczącą i część walidacyjną. W oparciu o ten sposób można ujawnić w trakcie procesu uczenia tendencje sieci do przeuczenia. W takim przypadku wskazane jest, aby zmniejszyć ilość neuronów ukrytych i/lub liczbę warstw ukrytych. W sytuacji przeciwnej, gdy sieć nie posiada dostatecznych możliwości do modelowania rzeczywistej funkcji, przeuczenie nie pojawia się, ale wtedy mimo długiego czasu uczenia błąd uczenia, ani błąd walidacyjny nie spadnie do satysfakcjonującego poziomu (Hertz, Krogh, Palmer, 1995).

W celu zwiększenia poziomu zaufania do ostatecznego modelu zwykle praktykuje się wydzielenie trzeciego zbioru przypadków - tak zwanego zbioru testowego. Ostateczna postać modelu (sieci) sprawdzana jest za pomocą zbioru testowego (zbiór ten używany jest tylko raz).

Kolejnym istotnym parametrem jest liczba „epok" uczenia sieci (jedna prezentacja podzbioru próbek uczących wraz z odpowiednią korekcją wag). Liczba próbek podzbioru jest nazywana rozmiarem epoki. Jeżeli będzie to liczba zbyt mała to sieć nie zdąży się nauczyć problemu. Zbyt wielka liczba epok z kolei doprowadzi do znacznego wydłużenia czasu uczenia się sieci i zwiększy ryzyko jej przeuczenia. Parametr ten

stosunkowo łatwo ustalić eksperymentalnie. Obserwując wykres uczenia się sieci widzimy, że po przekroczeniu pewnej liczby epok proces ten praktycznie nie postępuje [8].

Następnym parametrem jest określenie dopuszczalnego błędu, czyli żądanej jakości dopasowania sieci do danych empirycznych. Najczęściej jakość mierzy się przy pomocy średniego błędu kwadratowego (RMS). W niektórych typach sieci stosuje się również inne miary np.: SSE (Sum squared error performance function), MAE (Mean absolute error) i inne.

Po osiągnięciu tej wartości algorytm uczenia sieci zostanie przerwany. Gdy ustalimy ten parametr na zbyt niskim poziomie, sieć zostanie słabo wyuczona, natomiast gdy na zbyt wysokim – może go nigdy nie osiągnąć. Nawet gdybyśmy kontynuowali naukę sieci, zwiększając liczbę epok, to nie ma ona szans osiągnąć założonego pułapu. Często w analizie szeregów czasowych MSE ustala się na poziomie 1% przeciętnej wartości badanej cechy.

Kolejnym parametrem jest krok uczenia, który dotyczy wielkości zmian wag neuronów w każdym kolejnym cyklu uczenia. Zbyt duży krok spowoduje problemy ze zbieżnością sieci. Zbyt mały krok spowoduje olbrzymie straty czasu i nadwrażliwość sieci [6].

Jeszcze jednym ważnym parametrem sieci jest czas jej uczenia. Czas potrzebny na uczenie sieci rośnie wykładniczo wraz ze wzrostem liczby danych wejściowych. Przy stałej liczbie wejść rośnie wykładniczo wraz z ze wzrostem ilości neuronów. Natomiast rośnie liniowo wraz ze wzrostem liczby epok uczenia. Bardzo często przy obecnych algorytmach i ograniczonej mocy komputerów, czas budowy rozsądnej sieci może przewyższyć wielokrotnie cały dostępny czas, po upływie którego, sieć jest już niepotrzebna. Znany jest klasyczny paradoks, że „zrobienie prognozy na jutro może potrwać trzy dni".

Kluczowym problemem we wszystkich badaniach ilościowych jest dostępność i jakość danych. W zasadzie nie ma możliwości zdobycia wszystkich potrzebnych danych o odpowiedniej jakości. Na problem ilości danych prawie nigdy nie mamy wpływu, gdyż jest on ograniczony kosztami, możliwościami technicznymi.

Innym problemem przy konstrukcji sieci neuronowej jest to, że sieci neuronowe przetwarzają wyłącznie dane numeryczne należące do ściśle określonego przedziału. Stwarza to problemy w tych sytuacjach, gdy dane należą do innego przedziału, jeśli występują braki danych, lub też, gdy dane mają charakter nienumeryczny. Dane numeryczne są przeskalowywane do właściwego dla sieci przedziału, przy czym może to być wykonane automatycznie lub sterowane przez użytkownika.

Określenie rozmiaru zbioru uczącego, czyli właściwej liczby przypadków wymaganych do nauczenia sieci neuronowej, stwarza w ogólności istotne problemy. Istnieją pewne reguły heurystyczne, które uzależniają liczbę wymaganych przypadków od rozmiaru sieci. Najprostsza z nich mówi, że liczba przypadków powinna być dziesięciokrotnie większa od liczby połączeń występujących w sieci. W rzeczywistości liczba potrzebnych przypadków jest również uzależniona od złożoności funkcyjnej poddanej modelowaniu i praktycznie rośnie ona nieliniowo, co powoduje, że nawet przy dość małej liczbie zmiennych liczba ta jest ogromna. Problem ten w literaturze nazywany jest jako „przekleństwo wielowymiarowości". W przypadku posiadania mniejszych zbiorów danych należy zdawać sobie sprawę z faktu, że zmuszeni jesteśmy używać sieci niedostatecznie nauczonej. Zazwyczaj w takich przypadkach najlepszym sposobem postępowania będzie dopasowanie modelu liniowego [8].

Rozwiązanie wielu rzeczywistych problemów bywa także utrudnione z uwagi na niepełnowartościowość danych tworzących zbiór uczący. Ta niepełnowartościowość przejawia się tym, że odpowiednie dane są w odpowiednim zbiorze, ale nie wnoszą tak wiele informacji, jak powinny. Przyczyny tego mogą być różne. Na przykład wartości pewnych zmiennych mogą być zniekształcone przez występujące szumy albo też niektóre zestawy danych mogą być niekompletne. Powszechnie sądzi się również, że sieci neuronowe są odporne na szumy. Oczywiście jest tak w rzeczywistości, jednak odporność ta ma swoje granice.

Dane można skalować, standaryzować lub normalizować według wielu algorytmów. Techniki te stanowią problem sam w sobie z punktu widzenia zmiany parametrów statystycznych danych przetworzonych. Najczęściej stosowane jest zwykłe przeskalowanie danych do zadanego przedziału.

3. Omówienie wyników predykcji sieci neuronowych dla wybranych wskaźników na przykładzie Warszawskiej Giełdy Papierów Wartościowych

Jednym z istotnych zastosowań sieci neuronowych jest predykcja zachowań giełdy papierów wartościowych.

Celem badań było znalezienie odpowiedzi na pytanie czy rynek akcji zachowuje się w przypadkowy sposób, nie posiadając właściwie przewidywalnych trendów, czy też trendy te mogą być prognozowane.

Zachowanie giełdy ma charakter prognozowalny, jeśli istnieje funkcja, która opisuje badany rynek i która przy pomocy sieci może zostać zidentyfikowana.

W celu uzyskania odpowiedzi na to pytanie zostały skonstruowane odpowiednie sieci neuronowe w oparciu

o pakiet programu Qnet 2000 [1], które następnie poddano trenowaniu i testowaniu. Wyniki zostały przedstawione i omówione w bieżącym rozdziale.

Budując model do celów prognostycznych możemy zastosować dwa podejścia. Pierwsze to podejście autoregresyjne, które oznacza, że badany proces jest modelowany na podstawie jego własnych opóźnień. W tym przypadku prognozujemy np. wskaźnik WIG jedynie na podstawie jego przeszłych wartości. Drugie podejście to wprowadzenie do modelowania danych przekształconych np. wartości wskaźników technicznych. W pracy zastosowano drugą metodologię.

Horyzont czasowy prognozy może być dowolny i różnie określony. Możemy robić prognozy wartości np. WIG'u na następny tydzień lub na następne 5 sesji. Różnica jest istotna. W pierwszym przypadku wyznaczamy średnie wartości WIG dla każdego tygodnia. W drugim przypadku wyznaczamy średnie 5-cio sesyjne. Należy zwrócić uwagę, że posługując się średnimi możemy znacznie skrócić liczbę dostępnych próbek uczących i testowych. Gdybyśmy szacowali średnie tygodniowe to próbek będzie tyle co tygodni a więc liczba danych zmniejszy się pięciokrotnie. Przy średnich 5-cio sesyjnych strata będzie równa czterem sesjom, a więc nieistotnie mało.

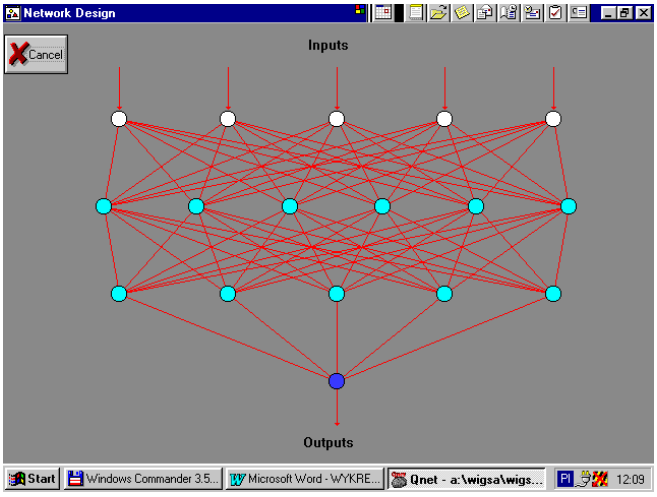
Drugim, obok prognoz kierunku zmian, jest problem prognozowania wartości badanego waloru. Jest to uzupełnienie powyższych analiz o wskaźnik skali zmian. Staramy się nie tylko ustalić kierunek zmian ale i ich wielkość. Jest to zadanie dużo trudniejsze ze względu na wymaganą wysoką precyzję.

Z oczywistych względów chcielibyśmy robić prognozy na jak najdalszy moment w przyszłości. Trzeba sobie jednak zdawać sprawę z tego, że błędy prognoz rosną gwałtownie wraz ze zwiększaniem horyzontu prognozy.

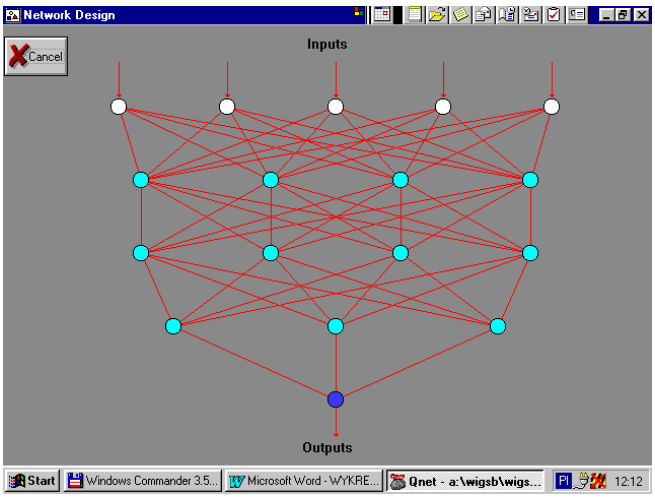
Klasyczne podejście do budowy sieci nie pozwala zbyt dokładnie kontrolować zdolności generalizacyjnych sieci. Trenujemy sieć na próbie uczącej, następnie sprawdzamy jej zdolności na małej próbie testowej. Wynik tego sprawdzianu nas zadowala lub nie. Jeżeli nie, to zaczynamy budowę i uczenie sieci od nowa. Dla małych sieci jest to strategia do przyjęcia - dla dużych nie.

Konstrukcję sieci rozpoczęto od zbadania wpływu wartości wag na starcie na zdolności generalizacyjne sieci. Uruchamiano sieć z ustaloną topologią z różnymi wagami (wyznaczonymi z funkcji Random). Uzyskane rezultaty pozwalają stwierdzić, że początkowe wagi nie wpływają w sposób istotny na końcowy wynik działania sieci. Następnie zbadano wpływ ilości neuronów oraz ilości warstw na jakość pracy sieci. Wyniki dla odmiennie skonstruowanych sieci, trenowanych na przekształconych danych historycznych WIG, zebrano w tabeli 1.

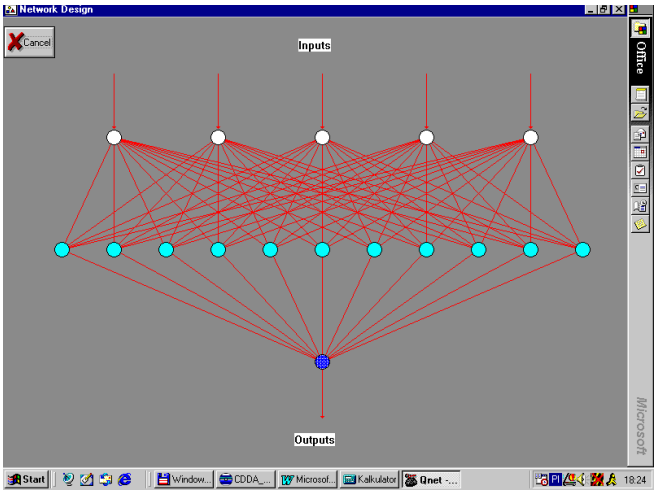
Sieć A to sieć z dwoma warstwami ukrytymi (Rys.1) z łączną ilością 11 neuronów w tych warstwach.



Rys.1 Topologia sieci A.

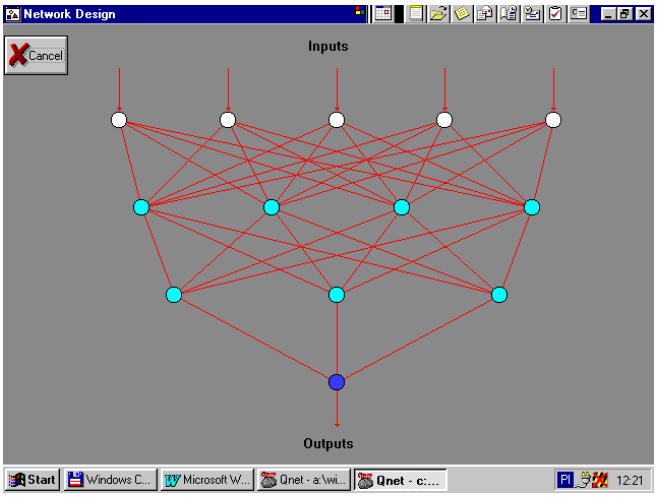


Rys.2 Topologia sieci B.



Rys.3 Topologia sieci C.

Podobną ilość neuronów (Rys.2,3) w warstwach ukrytych ma sieć B i C, odpowiednio z trzema i z jedną warstwą ukrytą.



Rys.4 Topologia sieci D.

Sieć D jest siecią z dwoma warstwami ukrytymi z łączną liczbą trzynastu neuronów. Porównując otrzymane predykcje dla różnych sieci, uwzględniając współczynnik tolerancji 1% oraz 0.5% (dopuszczalna różnica pomiędzy prognozowaną względną wartością a jej rzeczywistą względną wielkością) można zauważyć, że przy większym współczynniku tolerancji najlepszą siecią jest sieć z dwoma warstwami ukrytymi, z trzema i czterema neuronami w tych warstwach (sieć D). Z przeprowadzonych testów wynika, iż na każde pięć przypadków testowych, sieć udzieliła pięciu poprawnych odpowiedzi (mieszczących się w granicach założonego błędu).

RODZAJ SIECI	LICZBA DOBRYCH / ZLYCH PREDYKCJI	
	WSPÓLCZYNNIK TOLERANCJI 0.01	WSPÓLCZYNNIK TOLERANCJI 0.005
A	4/1	2/3
B	5/0	2/3
C	4/1	2/3
D	5/0	2/3

Tab.1.Wyniki prognozy dla wskaźnika WIG przy różnej topologii sieci.

Podobną liczbę poprawnych predykcji można uzyskać przy większej ilości neuronów i większej liczbie warstw ukrytych (sieć B), lecz ze względu na bardziej złożoną strukturę, a tym samym wydłużony czas uczenia, jest to sieć mniej atrakcyjna. Sieć A daje gorsze rezultaty dla danych testujących, natomiast C jest gorsza w procesie uczenia (wolniej dochodzi do stabilnych wyników pracy).

Przy współczynniku tolerancji 0.5%, stawiającym w stosunku do sieci dużo większe wymagania (problem ten będzie jeszcze dokładniej omawiany), wyniki uzyskane przy pomocy analizowanych sieci są identyczne, jednakże procent poprawnych odpowiedzi obniżył się i wynosi 40% w każdym przypadku.

Analiza sieci A z większą ilością neuronów w warstwach ukrytych w stosunku do sieci D wykazuje, że dla dodatkowych neuronów zarówno w warstwie pierwszej ukrytej jak i drugiej, procentowy udział ich wag jest dużo niższy niż pozostałych neuronów wbudowanych w te warstwy.

SIEĆ	UDZIAŁ PROCENTOWY POSZCZEGÓLNYCH NEURONÓW W WARSTWIE WEJŚCIOWEJ					UDZIAŁ PROCENTOWY WAG POSZCZEGÓLNYCH NEURONÓW W KOLEJNYCH WARSTWACH UKRYTYCH													
						1 W					2W						3W		
	1N	2N	3N	4N	5N	1N	2N	3N	4N	5N	1N	2N	3N	4N	5N	6N	1N	2N	3N
A	43	46	10	1	0	18	40	2	18	22	16	13	9	16	29	17	-	-	-
B	44	45	10	1	0	24	30	16	30	-	13	35	19	33	-	-	48	44	8
C	45	50	10	5	0	10	5	6	12	20	3	8	7	18	5	6	-	-	-
D	47	43	10	0	0	21	21	33	25	-	20	70	10	-	-	-	47	45	8

Tab.2. Udział procentowy poszczególnych neuronów w warstwie wejściowej oraz w kolejnych warstwach ukrytych w zależności od rodzaju sieci.

Należy zauważyć, że udział procentowy wag poszczególnych neuronów w warstwie wejściowej jest bardzo podobny dla wszystkich wariantów sieci. Istotnym jest spostrzeżenie, iż decydujące znaczenie mają średnie kroczące względne, 5 i 10 –sesyjnej (zawarte w dwóch pierwszych kolumnach danych wejściowych), natomiast niewielki jest wpływ średniej kroczącej względnej 200-sesyjnej (piąta kolumna danych).

Proces uczenia sieci jak i testowania kończony był w momencie uzyskania rezultatów, które były stabilne w stosunkowo długim okresie pracy sieci. Współczynniki korelacji jak i błędy RMS są bardzo zbliżone dla wszystkich typów sieci.

Na podstawie powyżej zamieszczonej analizy wybrano do dalszych badań sieć o strukturze D.

Wybór ten jednocześnie jest potwierdzeniem reguły, że ostateczna postać sieci posiada dwie warstwy ukryte, które zawierają w każdej warstwie po około połowie sumy neuronów warstwy wejściowej i wyjściowej.

W ten sposób skonstruowana sieć posłużyła do predykcji trzech odmiennie wyznaczonych indeksów Warszawskiej Giełdy Papierów Wartościowych (WGPW). Do analizy wybrano dane dotyczące wartości WIG'u (parametru opisującego zachowanie wszystkich spółek notowanych na WGPW) oraz WIG20 dla 20 największych spółek notowanych na giełdzie z ostatnich sześciu lat, a także ceny wszystkich spółek (138) wchodzących w skład WGPW (dane oznaczono jako WIGS).

Dane te zostały przekształcone w serię prostych wskaźników technicznych wykorzystujących cenę zamknięcia. Zostało wprowadzonych pięć wskaźników dla każdej spółki giełdowej, WIG'u oraz WIG20 - czyli pięć neuronów wejściowych, średnie kroczące 5, 10, 20, 50 i 200-sesyjne. Średnie te zostały następnie podzielone przez odpowiednie ceny (wartości) zamknięcia na dany dzień. Dane te (pięć względnych średnich kroczących, jako dane wejściowe oraz wskaźnik ROC, jako wyjście sieci). Zweryfikowane zostało założenie, iż dane te pozwalają na określenie przyszłych wartości badanych zmiennych, a więc i trendów (rosnących lub malejących).

W procesie przygotowania danych były wykorzystane pojęcia: średniej ruchomej i impetu.

Średnie ruchome (zwane również średnimi kroczącymi) należą do klasycznych narzędzi stosowanych do analizy technicznej. Ich popularność znacznie wykracza poza zagadnienia analizy szeregów finansowych, gdyż są one z powodzeniem stosowane w różnorodnych analizach danych. Średnie tego typu ułatwiają identyfikację trendów, pozwalają na filtrację danych, wskazują na punkty zwrotne w analizowanych szeregach.

Średnie ruchome są szeregiem uśrednionych wartości szeregu pierwotnego, przy czym proces uśredniania realizowany jest na podstawie danych mieszczących się we fragmencie szeregu o określonej długości (podszereg uwzględniony w czasie obliczeń nazywany jest oknem). Określenie średnia ruchoma lub krocząca związane jest z tym, że okno wyznaczające uśredniane wartości podąża za bieżącą wartością szeregu.

Wyznaczanie szeregu wartości średnich ruchomych jest formą filtracji danych mającej na celu usunięcie bądź zmniejszenie wpływu wahań krótkookresowych (średnia ruchoma jest więc rodzajem filtru). Uzyskany szereg wartości uśrednionych reprezentuje zmiany będące wynikiem wahań o dłuższych okresach.

Wraz z wydłużaniem szerokości okna zastosowanego do obliczania średniej wpływ wygładzania (eliminacji wahań krótkookresowych) jest coraz mocniejszy i uzyskane rezultaty obliczeń informują w coraz większym stopniu o tendencjach długookresowych. Dobór szerokości okna jest uzależniony od celu analizy.

Ze względu na sposób obliczania średniej możliwe jest wyróżnienie różnych typów średnich ruchomych. W pracy zostanie zastosowana prosta średnia ruchoma.

Prosta średnia ruchoma jest średnią arytmetyczną ze wszystkich wartości wchodzących w skład przyjętego okna.

Sposób jej wyznaczania przedstawia wzór

$$S^{pr}_{t} = 1/n \sum_{i=t-n+1}^{t} x_i$$

gdzie xi oznaczają ceny w czasie i.

Pojęcie impetu definiuje się jako tempo zmiany cen. Analiza impetu nie uwzględnia bezwzględnych wartości cen, lecz zajmuje się wyłącznie zmianami zachodzącymi w określonym okresie czasu. Impet (zwany również pędem lub momentum) wyrażany jest w sposób ilościowy przy pomocy wskaźników impetu. Pojęcie impetu jest zbliżone do pojęcia szybkości i wyrazić można przy pomocy ogólnej formuły:

$$M = \Delta p / t$$

gdzie:

M - impet, Δp zmiana ceny, t - czas.

Istnieje cały szereg miar impetu do najpopularniejszych można zaliczyć wskaźnik zmian (ROC - Rate of Change):

$$ROC = x_t / x_{t-n}.$$

Dane dla WIG oraz WIG20 zostały przekształcone zgodnie ze wzorami podanymi powyżej :

Wejścia:

5-cio sesyjna średnia ruchoma/aktualna wartość za-

mknięcia
10-cio sesyjna średnia ruchoma/aktualna wartość zamknięcia
20-sto sesyjna średnia ruchoma/aktualna wartość zamknięcia
50-cio sesyjna średnia ruchoma/aktualna wartość zamknięcia
200-tu sesyjna średnia ruchoma/aktualna wartość zamknięcia
Wyjście:

Zmiana względna wskaźnika po 5 sesjach od aktualnej wartości zamknięcia - wskaźnik zmian ROC (Rate of Change). W przypadku WIGS wejścia sieci stanowiły 5,10,20,50,200-sesyjne średnie obliczone względem cen akcji spółek z dnia 08-04-2002 roku dla wszystkich 138 spółek notowanych na WGPW. Wyjście stanowiły ROC cen poszczególnych spółek po 5-ciu sesjach od daty podanej wyżej.

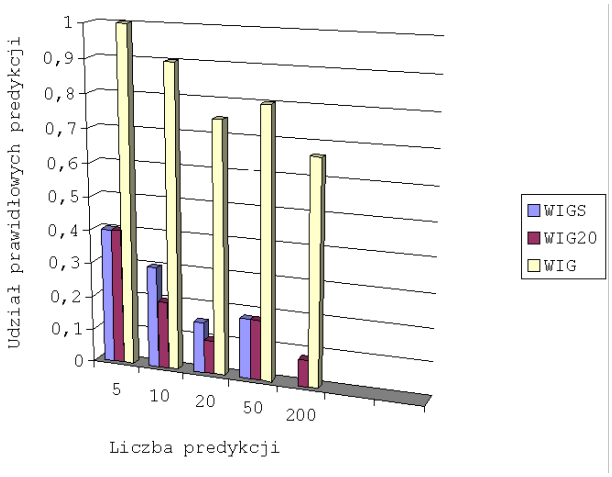
Nie korzystając z żadnego modelu można stwierdzić, że poziom wskaźnika WIG'u może zmienić swoją wartość co najwyżej o 10% w stosunku do obecnej wielkości. Ponieważ WIG, najczęściej, nie zmieniał się więcej niż o 4% to prognozę a priori można jeszcze zawęzić. Przeciętne zmiany WIG w badanym okresie wynosiły średnio 200 punktów z sesji na sesję (co do wartości bezwzględnej). Gdyby wyrazić te wartości w procentach byłoby to : 1.3%. Aby prognozy miały istotną wartość to błędy w dłuższych okresach powinny być mniejsze niż 1%. Nakładając na to doświadczenia inwestorów, można powiedzieć, że przyszła wartość WIG jest dość dobrze znana. Zadaniem sieci jest poprawianie tych przewidywań. W praktyce oznacza to, że prognozy nie powinny różnić się od rzeczywistości o więcej niż połowę tego, co i tak już wiemy. Wtedy trud włożony w budowę sieci może się opłacić. W praktyce oznacza to, że sieć powinna dawać prognozy ze średnim błędem nie przekraczającym, dla WIG-u, 1% bieżącej jego wartości.

Zaprojektowana sieć pozwala na uzyskanie lepszych rezultatów. Otrzymane wyniki zostały zaprezentowane dla współczynnika tolerancji 0.01 na Rys.5., a dla współczynnika tolerancji 0.005 na Rys.6

Analiza poniżej zamieszczonego wykresu wskazuje na doskonałą zdolność sieci do prognozowania kierunku i wartości względnej zmiany wskaźnika WIG'u (przy współczynniku tolerancji 0,01) w stosunku do pozostałych badanych wskaźników (WIG20 i WIGS).

Wysoką pewność prognozy wskaźnika WIG otrzymuje się dla predykcji pięciosesyjnych, maleje ona prawie monotonicznie i dla prognoz długoterminowych spada, jednak nadal jest ona duża (66% dla prognozy 200-tusesyjnej).

Generalizacja prognozy długoterminowej przy tym stopniu tolerancji jest możliwa jedynie dla WIG'u.

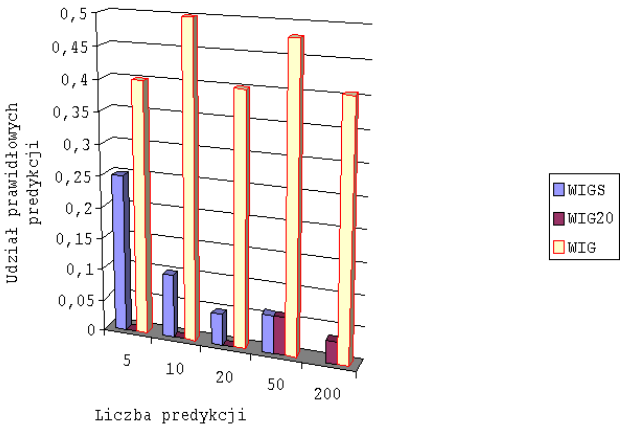


Rys.5 Udział prawidłowych predykcji w całkowitej liczbie predykcji (współczynnik tolerancji 0,01)

Prognozy wskaźników WIG20 i WIGS nie są zadowalające. Predykcje krótkoterminowe wyznaczone zostały przez sieć poniżej 50%, natomiast w długim horyzoncie czasowym spadają nawet poniżej 20%. W przypadku tych wskaźników wystąpił słaby proces generalizacji podczas procesu uczenia sieci. Przy założonym poziomie tolerancji (0.01) można stwierdzić, że użyteczność tych modeli jest niewielka.

Ceny akcji spółek wchodzących w skład WIG-u nie podlegają predykcji (WIGS), oznacza to, że rzeczywiście są one kształtowane losowo. Model neuronowy był w stanie zapamiętać zbiór uczący, ale nie potrafił reagować właściwie w stosunku do przypadków pochodzących ze zbioru testowego.

Prognozy krótko i długookresowe dla wskaźnika WIG20 przy współczynniku tolerancji 0.5%, (wypełnienie którego pozwala na dokonywanie korzystnych inwestycji), są bliskie zeru.

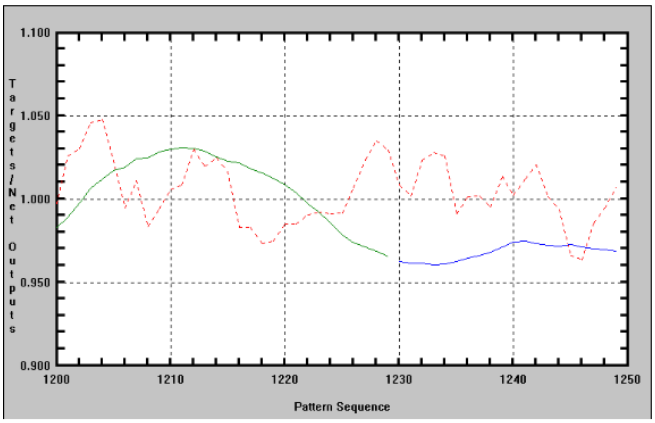


Rys.6 Udział prawidłowych predykcji w całkowitej liczbie predykcji (współczynnik tolerancji 0,005)

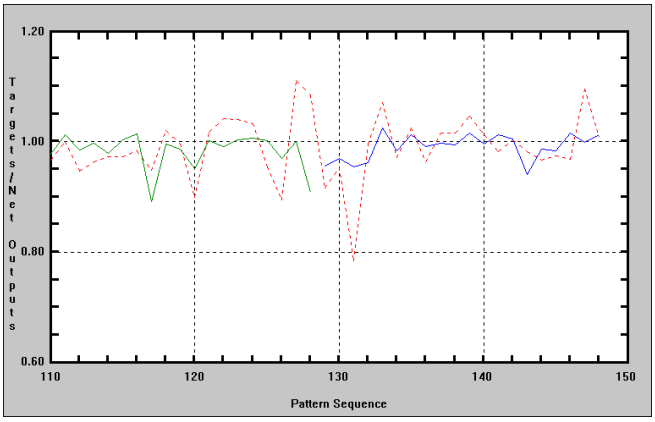
Udział udanych prognoz dla WIGS jest niewiele większy niż w przypadku WIG20, jedynie dla testu wykorzystującego pięć cen akcji spółek wynosi on nieco ponad 20%.

W przypadku WIG'u dobre predykcje utrzymują się na stałym, dość wysokim poziomie (od 40 - 50%) dla każdego z wybranych zbiorów testowych.

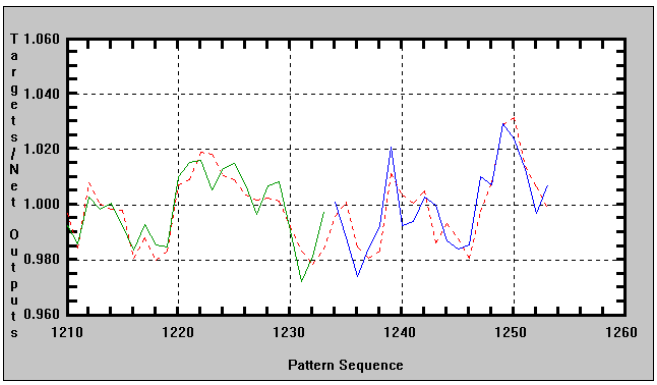
Z przeprowadzonych analiz wynika, iż relatywnie proste modele mogą dawać dobre wyniki w prognozowaniu indeksu wszystkich spółek WIG-u (Rys.9). Badania opublikowane wcześniej przeprowadzone przy użyciu odmiennie skonstruowanych sieci neuronowych potwierdzają otrzymane w pracy wyniki. Interesującą informacją jest brak możliwości predykcji dla indeksu 20-stu spółek (Rys.8) oraz wniosek, że nie można oczekiwać odpowiedzi poprawnej od sieci co do zachowania się cen (wzrostu lub spadku) dla wybranych spółek na podstawie zachowania się cen pozostałych (Rys.7).



Rys.7 Rysunek ilustrujący zależność względnego indeksu WIG20 dla predykcji 20-sto sesyjnej w procesie trenowania sieci (zielony), testowania sieci (niebieski) w stosunku do rzeczywistej względnej wartości (czewony).



Rys.8 Rysunek ilustrujący zależność względnego indeksu WIGS dla predykcji 20-sto sesyjnej w procesie trenowania sieci (zielony), testowania sieci (niebieski) w stosunku do rzeczywistej względnej wartości (czerwony).



Rys.9 Rysunek ilustrujący zależność względnego indeksu WIG dla predykcji 20-sto sesyjnej w procesie trenowania sieci (zielony), testowania sieci (niebieski) w stosunku do rzeczywistej względnej wartości (czerwony).

4. Podsumowanie

Sieci neuronowe są nowym narzędziem informatycznym umożliwiającym konstrukcję modeli nieliniowych rozwiązujących złożone i trudne do identyfikacji zadania klasyfikacyjne i regresyjne. Używanie sieci neuronowych w analizie złożonych danych jest szczególnie korzystne, ponieważ prowadzi do konstrukcji modeli opartych na właściwościach samych danych, a nie na arbitralnych hipotezach tworzącego model badacza.

Do reprezentatywnych przykładów problemów rozwiązywanych za pomocą sieci neuronowych należy prognozowanie giełdowe. Zmiany cen akcji są szeroko znanym przykładem zjawisk podlegających klasyfikacji i rozpoznawaniu. Zjawiska te charakteryzują się z jednej strony złożonością i wielowymiarowością, zaś z drugiej strony występowaniem, w niektórych okolicznościach, składowych mających przynajmniej charakter deterministyczny.

Wielu analityków stosujących analizę techniczną wykorzystuje w związku z tym sieci neuronowe do wyznaczania na przykład prognoz cen akcji na podstawie dużej liczby czynników, takich jak kształtowanie się w przeszłości cen innych akcji i różnych wskaźników ekonomicznych.

Sieci posiadają szereg cech, dzięki którym mogą stanowić przydatne narzędzie modelowania i prognozowania zjawisk społeczno - ekonomicznych. Celowość ich zastosowania wynika z pewnych własności charakteryzujących wspomniany typ zjawisk, jak również ze sposobu budowy i funkcjonowania modeli neuronowych. Jako uzasadnienie stosowania modeli neuronowych w modelowaniu prawidłowości występujących na rynkach finansowych można przyjąć następujące fakty [7]:

- znaczna ilość zjawisk rozpatrywanych na gruncie

finansów ma charakter nieliniowy, co stanowi podstawową przesłankę do tego, aby do ich modelowania stosować narzędzia przystosowane do opisu zależności nieliniowych. Jednym z narzędzi spełniających ten warunek są jednokierunkowe sieci neuronowe. Posiadają one zdolności do aproksymacji dowolnych zależności nieliniowych jak również charakteryzują się zdolnościami generalizującymi,

• proces budowy modelu neuronowego polega na eksploracji dostępnych zbiorów danych i oszacowaniu na tej podstawie modelu opisującego stwierdzone prawidłowości. Stosowanie modeli tego typu nie wymaga znajomości postaci funkcji opisującej istniejącą prawidłowość. W związku z tym modele neuronowe mogą znaleźć zastosowanie wszędzie tam, gdzie nie jest znane dokładne prawo opisujące kształtowanie się badanych zależności. Nie wyklucza to możliwości stosowania sieci w przypadkach, gdy znana jest postać formuł matematycznych opisujących badany aspekt rzeczywistości, ale wówczas nakłady związane z oszacowaniem modelu neuronowego mogą być wyższe niż nakłady niezbędne do obliczenia parametrów danego w postaci równania prawa,

• modele neuronowe mają charakter adaptacyjny. Mogą służyć do opisu zależności zmieniających się w czasie. W chwili pojawienia się nowych danych przeprowadzony może zostać proces douczenia sieci, co umożliwia uwzględnienie w tworzonym modelu informacji zawartych w najnowszych obserwacjach,

sieć neuronowa może być traktowana nie tylko jako mechanizm opisujący przebieg zjawiska i generujący przyszłe jego wartości. Daje ona możliwości przeprowadzania wszechstronnej analizy badanego fragmentu rzeczywistości. Podstawowe informacje o systemie uzyskać można poprzez zastosowanie analizy wrażliwości modelu. Pozwala ona na przedstawienie charakteru związku pomiędzy badaną wielkością o poszczególnymi wpływającymi na nią czynnikami,

• proces szacowania i wykorzystania modeli neuronowych może być realizowany współbieżnie w systemach wieloprocesorowych lub przez szereg komputerów połączonych w sieć komputerową. Taki sposób realizacji obliczeń neuronowych pozwala na znaczne skrócenie czasu potrzebnego na realizację niezbędnych działań.

Wyniki uzyskane przy pomocy sieci neuronowych mogą być wykorzystywane samodzielnie lub też mogą stanowić uzupełnienie rezultatów uzyskanych przy pomocy innych technik – na przykład klasycznych metod statystycznych.

W pracy przedstawiono wyniki badań nad zdolnościami sieci w prognozowaniu wybranych wskaźników dotyczących funkcjonowania Warszawskiej Giełdy Papierów Wartościowych. Dostarczają one dość obiecujących informacji. Udało się zbudować prognozy wartości zmian WIG'u o 100 % skuteczności w krótkim i o 66% skuteczności w długim horyzoncie czasu (dla współczynnika tolerancji 0,01). Prognozy wartości tego indeksu są więc zadowalające.

Oczywiście należy zdawać sobie sprawę, że pokazane tu ujęcie jest tylko jednym z bardzo wielu możliwych. Wyniki te nie są ostateczne, mają przede wszystkim zachęcić do dalszych badań. Istnieje wiele innych typów sieci, które należało by przetestować. Powinno się optymalizować także inne parametry sieci niż te wzięte pod uwagę w pracy. Należało by zoptymalizować wartości współczynników technicznych, sprawdzić wpływ informacji fundamentalnych na możliwości prognostyczne. Można by zbudować sieć rozpoznającą formacje techniczne – do wspomagania prognoz długoterminowych. Dróg poprawy prezentowanych wyników jest wiele.

Literatura (References)

[1]. Qnet 2000 V2K build 721 firmy Neural Network Modeling 2002, www.horyzont.eu/wydawnictwo/biuletyn/informatyka/zalacznik/qnet 2000

[2] E. M. Azoff, *Neural Network Time Series Forecasting of Financial Market.* Wiley & Sons, Chichester 1994.

[3] A. Beltratii, S. Margarita, P. Terna, *Neural Networks for Economic and Financial Modeling.* International Thompson Computer Press, London 1996.

[4] S. T. Gallant, *Neural Network Learning and Expert Systems.* MIT Press. Cambridge 1993.

[5] S. Goonatilake, P. Treleaven (red.), *Intelligent Systems for Finance and Business*. Wiley & Sons, Chichester. 1995.

[6] J. Hertz, A. Krogh, R. G. Palmer, *Wstęp do teorii obliczeń neuronowych*. Wydawnictwo Naukowo-Techniczne, Warszawa 1995.

[7] S. Shochen, G. Ariav, *Neural Networks for Decision Support: Problems and Opportunities*, Vol.11, Decision Support Systems 1994.

[8] R. Tadeusiewicz, *Elementarne wprowadzenie do sieci neuronowych z przykładowymi programami.* Akademicka Oficyna Wydawnicza, Warszawa 1998.

[9] J. S. Zirilli, *Financial Prediction Using Neural Networks.* International Thompson Computer Press, London 1996.

A neuro-modeling approach to implementing Intelligent harmonic monitoring system for electrical girds

Neuro-modelne podejście do realizacji inteligentnego systemu monitorowania harmonicznych dla sieci elektrycznych

Yuriy Varetsky

Wrocławska Wyższa Szkoła Informatyki Stosowanej
ul. Wejherowska 28, 54-239 Wrocław

Treść. Przedstawiono nową koncepcję inteligentnego systemu monitorowania harmonicznych. Do oceny wskaźników harmonicznych napięcia na szynach rozdzielni sieci dystrybucyjnej, zasilającej obciążenia nieliniowe, wykorzystano sztuczną sieć neuronową. Zaproponowano uczenie sieci neuronowej na podstawie zbioru danych, które otrzymano w wyniku symulacji w przestrzeni czasowej warunków pracy elektrycznej sieci dystrybucyjnej. Omówiono przykład opracowanego inteligentnego systemu monitorowania.

Słowa kluczowe: sieć neuronowa, modelowanie w przestrzeni czasowej, symulacja, elektryczna sieć dystrybucyjna, monitorowanie harmonicznych.

Abstract. A new concept of intelligent harmonic monitoring system for electrical distribution grid is presented. For estimation of voltage harmonic indices at the distribution grid substation buses supplying nonlinear loads an artificial neural network has been used. It was proposed to train the neural network on the data set obtained from simulation of the electrical distribution grid operating conditions in the time domain. An example of the carried out intelligent monitoring system is discussed.

Keywords: neural network, time domain modeling, simulation, electrical distribution grid, harmonic monitoring.

1. Introduction

The expression "intelligent system" is often used to denote any combination of the usage of artificial neural networks (ANN), expert systems, fuzzy logic systems and other technologies, such as genetic algorithms in particular. Unlike conventional approach, intelligent system does not require mathematical models of a real objects. As compared with human being, the computer with artificial intelligence can rapidly solve problems. Computer operates continuously "tirelessly". It is not influenced by emotions or other human drawbacks. These systems are constructed on mathematical relations, which inherit "intelligence" or "knowledge" from expert estimations or field observation data, presented, as a rule, in the form of input/output pairs.

In general, relations between input and output variables are known only approximately, and a lot of efforts must be applied to find acceptable approximate correspondences. Systems of neural networks are able to "learn" automatically approximated relations between inputs and outputs, bypassing the overcoming the problem of the size complexity of the task. These approximated relations are often more efficient than the obtained on the basis of physical description of the phenomenon. It happens since these relations usually connect real values of input and output variables (for instance, measurements data) and are free from assumptions of certain theories, based on some human prejudices. Besides, neural network does not require any information regarding dependences themselves or their efficiency, which is defined by the chosen law of phenomenon description. Due to large volume of input information, the approximation error can be gradually reduced. Theoretically neural network can be trained to provide exact correspondence between input and output data. In general, systems of neural networks are able to "study" dependences in preset volume of data and establish input-output relations, based exclusively on certain subset of data. Thus, it is expedient that subset data used for "training" represent complete set of data. Dependences which are not "seen" in the selected subset will not be "studied" by the neural network. It should be noted, that the same restriction concern conventional algorithms of regression and classification.

Harmonic monitoring has become an important tool for harmonic management in electrical distribution systems. An increasing number of electric distribution system service providers are interested in installing harmonic monitoring equipment to measure the harmonic voltage and current waveforms in their power system to detect and mitigate the harmonic distortion problems. This can be explained by the fact, that more and more loads with non-linear characteristics are used in distribution systems. On the other hand, practically all power systems are characterized by constant varying loads. Such variation can be of daily or long term character as well as of random character – i.e. depending on the operation requirements, and occur even several times a minute. So operating and maintenance staff of utility companies and commercial consumers can not identify the level of investments for preventive measures and

means to avoid possible harmful consequences of the problem. The key task in solving the problem is the implementation of an effective power quality monitoring system in the distribution systems.

2. Concept of the monitoring system

Monitoring power quality in distribution systems is performed to study varying quality indices during certain time interval. Depending on the specific features of a distribution system monitoring can be carried out continuously (based on stationary instruments), periodically (within certain intervals, for instance, once year) or as requires (in the process of connection of new powerful loads, compensating devices, etc.)

It should be noted, that according to the requirements of majority of actual standards harmonic monitoring should be performed at least during one week. It is expedient to choose the points of harmonics monitoring as close as possible to the equipment (consumers) sensitive to voltage distortion. It is important to have information regarding all changes of the distribution system configuration (connection/disconnection of capacitor bank, harmonics filters, buses sections, transformers etc.).

To provide electromagnetic compatibility in a distribution system the voltage distortion indices in "common point" should meet the requirements of the actual standard. Testing procedure is carried out on the basis of corresponding measurements. Measurements are performed by certified laboratories in accordance with standard technique and with a certified and specialized measuring instruments such as harmonic analyzers. It is obvious, that such measurements are performed occasionally, this is why information regarding real state of the system may not be available for a long period of time, hence, the network and consumers will experience negative (in case of non conformity with the requirements) influence of harmonics. Changes of operating conditions in distribution system require the application of special measuring instruments, which are seldom installed at the stations due to their high cost. Besides at tie-substations there is permanent operating staff for servicing these devices and for the analysis the measurements. The analysis of the problem shows that it is desirable to perform continuous monitoring of the system operating conditions. Monitoring performed with minimum instruments and telemetry allows the dispatcher to evaluate voltage distortion in control points and perform necessary steps to eliminate the problem.

In the literature one can find references to already known methods of adequate harmonic sources identification in conditions of incomplete provision of distribution system with measuring facilities. Solutions based on conventional approaches are suggested in [1,2]. Some approaches using methodology of artificial neural networks for determination of characteristics of harmonics sources in electric network are suggested in [3,4]. Research published in [3] presents structural neural network for identification of harmonic sources in distribution system equipped with a few stationary installed instruments for harmonic measurement.

The suggested concept of the monitoring system does not require harmonic analyzers in controlled points of the distribution system and the development of a special network for transfer measurements to observation point [5, 6]. Existing telemetry channels and measuring devices available at the distribution system substations can be used for this purpose. It have been suggested to use neural network for "recognition" (identification) of voltage distortion indices at the buses of distribution system substations on the basis of the operating condition parameters available in the point of observation through telemetry (corresponding active and reactive powers, currents, power factors etc.). Outline concept of the monitoring system is shown in Fig. 1. An important feature of the developed monitoring system is the possibility to perform observation in real a time over the change of operating conditions of a distribution system segment and not only over one substation. Besides, such monitoring does not impose rigid requirements regarding the accuracy of a measuring devices and allows for the definition of the voltage distortion indices on the basis of measurements available at tie-substations. After collection and analysis of a data obtained as a result of monitoring, the decision concerning the necessity of measurements by specialized harmonics analyzers on the problem buses can be made.

On its merits we deal with indirect measurements of voltage distortion indices, since their values are obtained not on the basis of analysis of voltage waveforms, but by means of "recognition" its relations between values of powers, currents, etc. measured in different points of distribution system derived by a neural network.

It is known that the spectrum of harmonics and their magnitudes in electric network depend on active and reactive powers of non-linear loads, operating conditions of reactive power compensation equipment. It is obvious that under continuous variations of these factors it is difficult to define functional dependence between the parameters and harmonic magnitudes in several points of the distribution system. So, the possibility to apply the intelligent system for the solution of the problem is very attractive idea, because it can identify the required relations.

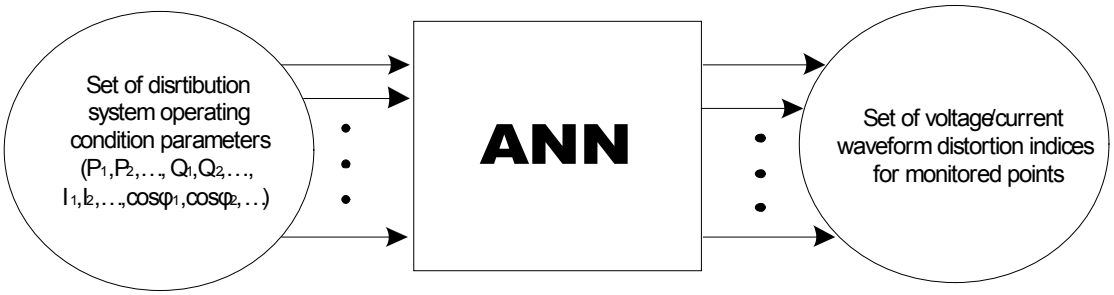


Fig. 1. Concept of the monitoring system

During the development procedure of such intelligent system, the most difficult problem is to obtain the data sets for ANN training. It is difficult to get needed data for the neural network training using field measurements. Problem is the collection of a data sets for all representative operating conditions during short time horizon. It was suggested to obtain these data from modeling the set of the representative operating conditions of the distribution system including non-linear loads in the time domain. The set of input and output pair obtained in such a way is the required training set for the neural network.

The next problem is the selection the neural network structure and acceptable method of its training. It is known that there are no exact criteria regarding the selection of neural network structure and the optimal method of its training. So, the commonly used approach has been chosen. In power engineering problems mainly the feed-forward neural networks are used. Their training is performed on the basis of training algorithm called "backpropagation". "Backpropagation" training algorithm is the method of iterative adjustment of weight coefficients until achieving the desired accuracy.

To obtain the best results, training set must correctly represent all expected changes in complete data set. That is why the correct choice of the load variations boundaries and electric network configuration changes are very important.

Majority of neural networks can be made very accurate (on the basis of training data) by means of increasing number of hidden layers and nodes in these layers. There are cases when the increase of independent variables makes the system more vulnerable to changes occurring in the output data. Thus, a large number of hidden layers and hidden nodes of the layer can make neural system more accurate for training data, but changes which will be presented in the following data (not included in training process) can be the reason of considerable shifts from expected result at the output. Thus it is necessary to find the compromise between the number of layers, nodes and the degree of accuracy, attainable with that training data.

3. Example of the intelligent monitoring system

A part of typical distribution grid including traction substations has been chosen. The grid substations are a source of harmonics and as a rule are supplied from a main line connected to 110 kV tie-substation bus. Supply of traction system is carried out through the 6 pulse diode bridge rectifier. Outline diagram of a monitoring system for the distribution grid is shown in Fig. 2.

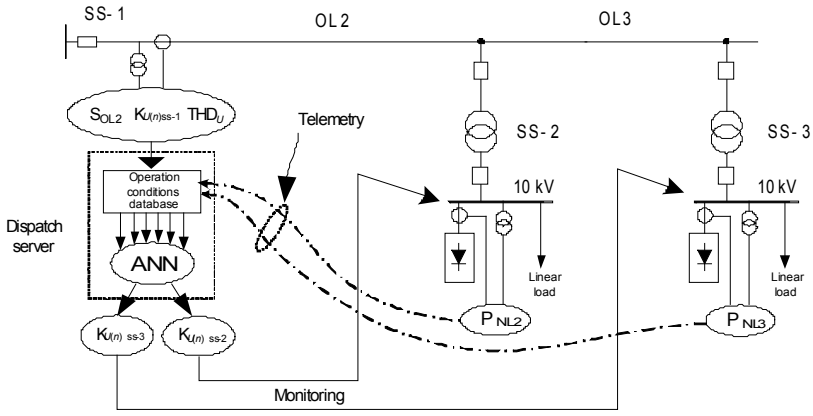


Fig. 2. Outline diagram of the monitoring system

Load of traction substation converters has a probabilistic character and can vary depending on the locomotive traffic schedule. Operating conditions both with practically zero loading of converters and with close to nominal are possible. The load can vary both continuously and by stepwise change (stop/start of the locomotive). The rectifier current harmonic magnitudes increase proportionally to the rectifier load. Substations in the investigated network are provided with standard measuring devices which are permanently installed on 10 kV buses of network substations.

Substations are not equipped with permanent de-

vices for harmonic measurements. In the distribution system the permanent harmonic analyzer is provided only at PL-2 at substation 1 where permanent operating staff performs on-line control of the given section of the electric grid. The dispatcher can observe the dynamics of substation load changes in time horizon by telemetries. Fig. 3 shows the artificial neural network (ANN) for the system of harmonic monitoring taking into account the transfer of telemetries from the substations SS-2 and SS-3 to substation 1.

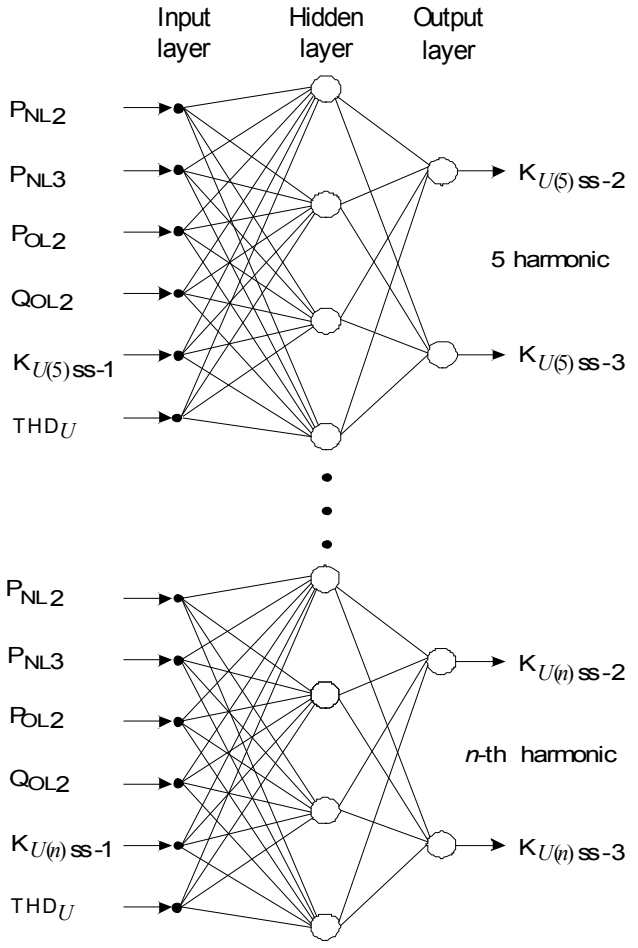


Fig. 3. The neural network architecture

Linear load of the substations changes in accordance with daily schedule of power consumption. Since the value of non-linear load is of stochastic character, there is no strict correspondence between the power of the system loads and variation of harmonics in the electrical grid.

Based on the experimental research for a given neural network the minimum number of necessary input signals (measurements) was defined to perform the monitoring task. The possibility of obtaining such measurements at substations of electric network and their transfer to control point at substation 1 is taken into account.

The neural network was created by means of subprogram NNTool from tools panel Neural Network of software MatLab. For neurons of the hidden layer, a transfer function in the form of hyperbolic tangent (TANSIG) is chosen. Function PURELINE is chosen as a transfer function for output layer neurons, since such a function can transfer any values in a wide range. Weight coefficients were adjusted to minimize complete root-square error between trained set of outputs and the set of real values. For training a neural network the function of scaled conjugate gradient (TRAINSCG) with back propagation of error is used.

Neural network was trained to evaluate coefficients of n-th voltage harmonic component at 10 kV buses of SS-2 та SS-3 (KU(n) ss-2, KU(n) ss-3) of the distribution grid. As it was pointed above, due to peculiarities of harmonic propagation calculations which allow for the analysis of the electrical grids separately for each harmonic. The neural network is divided into a number of parallel nets (individually for each harmonic) which have similar inputs and two outputs each, as it is shown in Fig. 3. Such approach allows for the improvement of the convergence and the increase of information memory. In the given case, only representative harmonics for the non-linear load of this distribution grid has been analyzed (5th, 7th, 11th etc).

In accordance with the proposed concept of intelligent monitoring system, simulations of the distribution system were performed to generate a training data set for the neural network. Modeling the distribution system for a variation of a load conditions in three-phase equivalent circuits in time domain has been carried out by means of MatLab tools (software Simulink). Voltage curves obtained as a result of computations for various load conditions have been decomposed in Fourier series. The voltage harmonic magnitudes characterize the corresponding values of powers chosen as inputs for the neural network. Set of operating conditions chosen for modeling comprised of a possible range of loads variations and combinations. Tab. 1 contains the fragment of the training data set for the neural network shown in Fig. 3. It illustrates the principle of generation of a training data set. Fig. 3. also contains testing data set which is not included to training data set. The testing data set is used for testing the neural network operation.

№	P_{NL2}, MW	P_{NL3}, MW	S_{OL2}, MVA	$K_{U(5)ss-1}$, %	$K_{U(5)ss-2}$, %	$K_{U(5)ss-3}$, %	K_U, %
1	1	0	5.4+j1.6	0,09	1,34	0,12	0,56
2	3	0	7.2+j2.2	0,25	3,8	0,34	0,99
3	5	0	8.9+j2.9	0,41	6,3	0,56	1,46
4	7	0	10.5+j3.7	0,56	8,47	0,75	1,78
5	0	1	5.5+j1.6	0,09	0,11	1,31	0,9
6	0	3	7.3+j2.1	0,26	0,34	3,85	1,61
7	0	5	9.0+j2.8	0,43	0,55	6,23	2,23
8	0	7	10.7+j3.6	0,58	0,74	8,61	2,66
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
21	1	7	11.6+j3.8	0,65	1,95	8,69	2,3
22	3	7	13.3+j4.5	0,82	4,52	8,94	2,11
23	5	7	15.0+j5.2	0,99	6,75	9,15	2,44
24	7	7	16.6+j6.0	1,13	8,8	9,28	3,53
Testing set							
25	2	5	10.8+j3.4	0,59	3,07	6,55	1,56
26	5	2	10.8+j3.4	0,58	6,14	3,21	1,71
27	4	6	13.3+j4.5	0,84	5,57	7,99	2,1
28	6	6	15.1+j5.3	0,97	7,15	7,68	2,22

Table 1. Example of training and testing data set for 5th harmonic

№	Real values		Training results		Absolute error	
	$K_{U(5)ss-2}$, %	$K_{U(5)ss-3}$, %	$K_{U(5)\ ss-2}$, %	$K_{U(5)\ ss-3}$, %	$\Delta K_{U(5)\ ss-2}$, %	$\Delta K_{U(5)ss-3}$, %
1	1,34	0,12	1,35	0,2	-0,01	-0,08
2	3,8	0,34	3,7	0,44	0,1	-0,1
3	6,3	0,56	6,17	0,65	0,13	-0,09
4	8,47	0,75	8,49	0,81	-0,02	-0,06
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
23	6,75	9,15	6,82	9,31	-0,07	-0,16
24	8,8	9,28	9,02	9,42	-0,22	-0,14
Results for testing set						
25	3,07	6,55	2,92	6,6	0,15	-0,05
26	6,14	3,21	6,27	3,11	-0,13	0,1
27	5,57	7,99	5,6	8,05	-0,03	-0,06
28	7,15	7,68	7,19	7,85	-0,04	-0,17

Table 2. Results of neural network training using the data of the set presented in Tab 1.2

Lines 25 – 28 in Tab. 2 show the results of evaluation of voltage waveform distortion indices at the buses of substations for data set that were not included in training set. Absolute error in Tab. 2 is calculated as the differences between real values obtained as a result of electrical grid operating condition simulations and data obtained as a result of neural network training.

Analysis of the results shows that the chosen structure of the neural network, both the set of input data and the step of change of training data pairs allow to provide desired accuracy of monitoring voltage distortion indices at the buses of the distribution grid substations.

4. Conclusions

1. A new concept of harmonic monitoring system for distribution grids was proposed. The concept allows for considerable reduction of the usage of expensive specialized instruments for monitoring harmonic sources in distribution systems.
2. The system enables monitoring voltage distortion indices in selected points of distribution system by the data of one or several stationary instruments for harmonics measurements and data available at lie-substations through existing telemetry channels (P, Q, U, I etc.).
3. The type of neural network for the developed of the monitoring system and the selection method of the training data sets is substantiated. The expediency of obtaining the training data sets for adjusting neural network through modeling of the number of operating conditions of the distribution grid is proved.

References (Literatura)

[1] J. E. Farach, M. V. Grady, A. Arapostathis, An optimal procedure placing sensors and estimating the locations of harmonic sources in power systems, IEEE Trans. Power Delivery 8, 3 (1993), 1303–1310.
[2] M. Najjar, G. T. Heydt, A hybrid nonlinear least squares estimation of harmonic signal levels in power systems, IEEE Trans. Power Delivery 6, 1 (1991), 282 – 288.
[3] R. K. Hartana, G. G. Richards, Constrained neural network-based identification of harmonic sources, EEE Trans. on Ind. Application 29, 1 (1993), 202 – 208.
[4] R. K. Hong, Y. C. Chen, Application of algorithms and artificial-intelligence approach fo locating multiple harmonics in distribution systems, IEEE Trans. Industry Appl. 29, 1 (1993), 202 – 208.
[5] Method of harmonic monitoring in distribution system: Pat. 35180 Ukraine. МПК7 G01R 23/16/ Y. Varetsky, T. Nakonechny. № u200802024; Publ. 10.09.2008. Bull. № 17.(ukr.)
[6] Y. Varetsky, T. Nakonechny, Monitoring Harmonic Sources in Distribution System by Neural Network Estimator // Proc. of 9 Int. Conf. „Electric power quality and utilization”. Barcelona, 9-10 October, 2007. www.leonardo_energy.org/archive/all/2007 Paper 1219, 4 P. 2007.

Spintronika

Spintronics

Tadeusz Mydlarz

Wrocławska Wyższa Szkoła Informatyki
Stosowanej
ul. Wejherowska 28, 54-239 Wrocław

Treść. W pracy pokazano przegląd wiedzy na temat mikro i spin elektroniki. Rozwój mikroelektroniki związany jest z fizyką półprzewodników. Poznanie własności fizycznych półprzewodników ich domieszkowanie i manipulowanie ładunkiem doprowadziło do rozwoju mikroelektroniki, powstawaniu zminiaturyzowanych podzespołów i urządzeń elektronicznych (diod i tnanzystorów, układów scalonych, mikroprocesorów, komputerów, telefonów komórkowych itp.). Pytanie jest zasadnicze „Czy manipulowanie ładunkiem w półprzewodnikach jest ograniczone jeśli tak to co dalej?" z rozwojem budową zminiaturyzowanych urządzeń elektronicznych nowoczesnych szybkich komputerów. Autor przedstawił krótki opis wiedzy i możliwości nowych materiałów magnetycznych mogących mieć ogromne zastosowanie w spinelektronice, nowej gałęzi wiedzy mającej początek w ostatnich latach ubiegłego wieku, w której to zasadniczą rolę odgrywa spin elektronu a właściwie manipulowanie nim.

Słowa kluczowe: połprzewodnik, półprzewodnik magnetyczny, spin elektronu, magnetoopór, nanostruktury, grafen, głowice indukcyjne, głowice rezystyrowe, urządzenia magnetorezystywne, twardy dysk.

Abstract. In the paper an overview of knowledge about microelectronics and spinelectronics is presented. The development of microelectronics is associated with the physics of semiconductors. Understanding the physical properties of semiconductors, their admixturing and manipulating of the charge has led to the development of the microelectronics and to the creating of the miniaturized components and electronic instruments and devices (diodes and transistors, integrated circuits, microprocessors, computers, mobile phones etc). The main question is: Is the manipulating of the charge in the semiconductors limited and if it is, what is next with the development and the construction of the miniaturized electronic devices and modern and fast computers? The author has presented a brief description of the knowledge and the possibilities of the new magnetic materials, that may have great applications in spinelectronics, which is a new branch of knowledge having the beginning in the last years of the previous century, and where the spin of the electron and the process of manipulation it plays the main role.

Keywords: semiconductors, diluted magnetic semicomductors, electron spin, magnetic resistivity, nanostructure, graphene materials, induction heat, resistivity heat, magnetoresistive device, hard dysk.

1. Wstęp

Spintronika (elektronika spinowa) w dzisiejszych czasach to bardzo modny kierunek rozwoju fizyki. Dotychczas rozwój mikroelektroniki i wszystkich urządzeń oparty był na manipulowaniu ładunkiem elektrycznym w materiałach półprzewodnikowych. Odkrycie i poznanie własności fizycznych półprzewodników typu n i p pozwoliło na powstanie wielu zminiaturyzowanych urządzeń elektronicznych układów scalonych, mikroprocesorów, złącz półprzewodnikowych p-n tranzystorów MOSFET, diod świecących. Wszystkie te podzespoły mają zastosowanie w budowie komputerów i wszelkiego rodzaju sprzętu elektronicznego. Poszukujemy coraz to nowszych materiałów magnetycznych, które można by było wykorzystywać do budowy twardych dysków, głowic, elementów pamięci. Materiały te powinny mieć co najmniej pokojową temperaturę uporządkowania magnetycznego Tc. W dzisiejszych komputerach w głowicach wykorzystuje się materiały wykazujące bardzo duży magnetoopór (opór elektryczny zależny od pola magnetycznego). Pytanie jest zasadnicze, czy istnieje granica manipulowania ładunkiem elektrycznym w półprzewodniku przez wprowadzanie atomów domieszkowych do czystego półprzewodnika, a przez to otrzymywanie coraz to ciekawszych nowych materiałów. Wydaje się, że ten czas nadchodzi. Uczeni wpadli na pomysł wykorzystania spinu elektronu w materiałach półprzewodnikowych, jego manipulowaniem, co pozwoliło na poznanie wielu nowych materiałów magnetycznych. Uporządkowanie magnetyczne w ciele stałym było przypisywane atomom obdarzonym trwałym momentem magnetycznym (metalom przejściowym 3d , żelazo, kobalt, nikiel z do końca nie zapełnionymi pod powłokami 3d), metalom ziem rzadkich (w pewnym zakresie temperatur). Atomy Fe, Co, Ni w określonej strukturze krystalograficznej oddziaływały wzajemnie ze sobą bezpośrednio siłami wymiennymi dając spontaniczne namagnesowanie poniżej temperatury Tc. Oddziaływania wymienne między atomami Ziem Rzadkich były pośrednie poprzez spolaryzowane elektrony przewodnictwa (oddziaływanie RKKY). W Układzie Mendelejewa jest bardzo mało pierwiastków wykazujących magnetyczne uporządkowanie. Jednak

połączenia różnych pierwiastków w związkach międzymetalicznych prowadzi nie jednokrotnie to uporządkowania magnetycznego i to czasem niespodziewanego [1]. O ile w związkach międzymetalicznych zawierających atomy obdarzone trwałym momentem magnetycznym jest to spodziewane i oczekiwane to w związkach aktynowca z niemetalem czy wodorem było dużym zaskoczeniem uczonych w czasach gdy te własności odkrywali [2]. Własności magnetyczne przypisywano ciału stałemu o określonej strukturze krystalicznej, ciału masywnemu. Dzisiaj uporządkowanie magnetyczne spotykamy w nanostrukturach kryształach molekularnych, nanorurkach a ostatnio w grafenie [3]. Grafen jedna z alotropowych odmian węgla odkryta w 2004 roku przez uczonych rosyjskich i brytyjskich (Andriej Gejm, Konstantin Nowosiołow [4]) jest zbudowany jest z pojedynczej warstwy atomów węgla tworzących połączone pierścienie sześcioczłonowe (dwuwymiarową siatkę o sześciokątnych oczkach, której struktura przypomina plaster miodu).

Spintronika to alternatywa dla klasycznej elektroniki gdzie zasadniczą rolę odgrywa spin elektronu [5]. Od wielu lat myśli się o wykorzystaniu spinu jako dodatkowego stopnia swobody w urządzeniach elektronicznych. Mam tu na myśli takie urządzenia jak tranzystor spinowy, spinowe procesory. Specjaliści zajmujący się różnymi dziedzinami fizyki, elektroniki i teorii informacji, informatycy stawiają sobie różne cele. Informatycy uważają, że manipulowaniem spinem daje możliwość zbudowania nowego typu procesora logicznego związanego z kwantowym charakterem spinu.

2. Spin elektronu a zakaz Pauliego

Spin elektronu to w 100% wielkość kwantowa. Ma ona reputacje wielkości, której nie można zrozumieć. Analogia do klasycznego momentu pędu wokół własnej osi jest wysoce niedoskonała. Np. cząstka nie może stracić ani zyskać spinu, może jedynie zmienić jego kierunek. Fermiony mają spin połówkowy (2n+1)1/2 (elektron, neutron, proton) a bozony (foton, pion) maja spin całkowity. Spin elektronu jest skwantowany i jego wartość wynosi:

$S = \hbar(s(s+1))1/2$

Gdzie $s = ½$ jest spinowa liczbą kwantową.

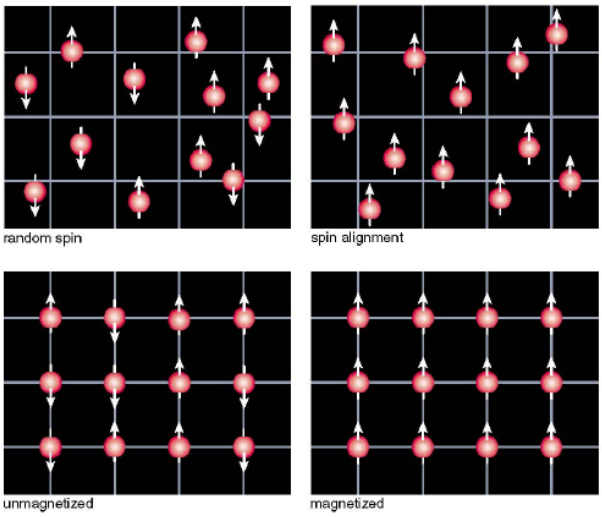
Rzut spinu na oś z (np. kierunek przyłożonego pola magnetycznego) jest skwantowany

$Sz = m_{sh}$

Gdzie *ms* przyjmuje wartośći + ½ i – ½

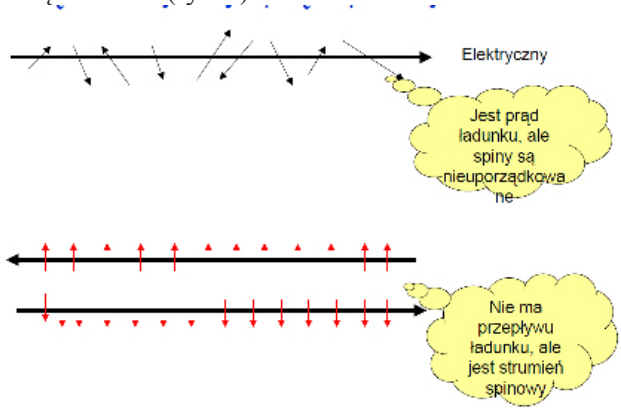
Omawiając oddziaływania spinowe musimy pamiętać o dwóch cechach spinu. Pierwszą cechą jest to, że związanemu z nim momentowi pędu odpowiada moment magnetyczny, a ten ulega wpływowi pola magnetycznego. Druga cecha to połówkowość spinu, która oznacza, że obsadzanie stanów elektronowych przebiega ściśle według zasady Pauliego, zabraniającej obsadzania tego samego stanu przez dwa elektrony. Jeśli jeden stan jest już obsadzony przez elektron o określonej orientacji spinu to drugi elektron musi mieć spin przeciwny. Jest to wystarczający powód aby wykorzystać spin jako element logiczny.

Spiny mogą być różnie uporządkowane (rys. 1)



Rysunek 1. Różne rodzaje uporządkowania spinów

Może wystąpić przepływ prądu gdy spiny są nieuporządkowane lub nie ma przepływu prądu a spiny są uporządkowane (rys. 2)

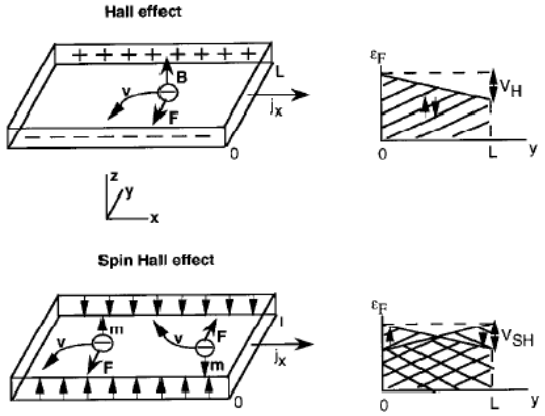


Rysunek 2. Możliwości przepływu prądu elektrycznego w przypadku nieuporządkowanych i uporządkowanych spinów

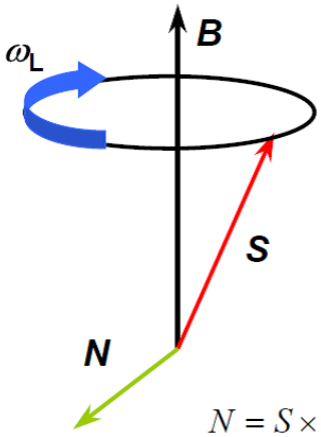
Często obserwuje się spinowy efekt Halla (rys. 3)

Na kierunek spinu możemy wpływać przez zastosowanie: pola magnetycznego, rozpraszania na domieszkach magnetycznych i optycznie.

W polu magnetycznym spin wykonuje precesję wokół kierunku pola z częstością Larmora $\omega_L = gB$ (rys. 4)



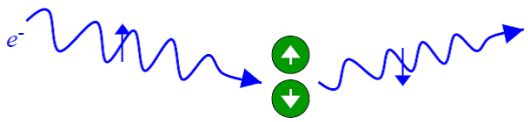
Rysunek 3. Spinowy Efekt Haala



$$N = S \times B$$

$$\frac{dS}{dt} = \gamma S \times B = S \times \omega_L$$

Rysunek 4. Precesja spinu w polu magnetycznym



Rysunek 5. Rozpraszanie spinu na domieszkach magnetycznych

Na rys.5 pokazano rozpraszanie spinu na domieszkach magnetycznych. Spinem można manipulować optycznie, naświetlając kwantem energii $E = h\sqrt{}$ *InMnAs* otrzymano uporządkowanie magnetyczne w tym związku rys. 6 [6].

W wielu materiałach opór elektryczny zależy od pola magnetycznego

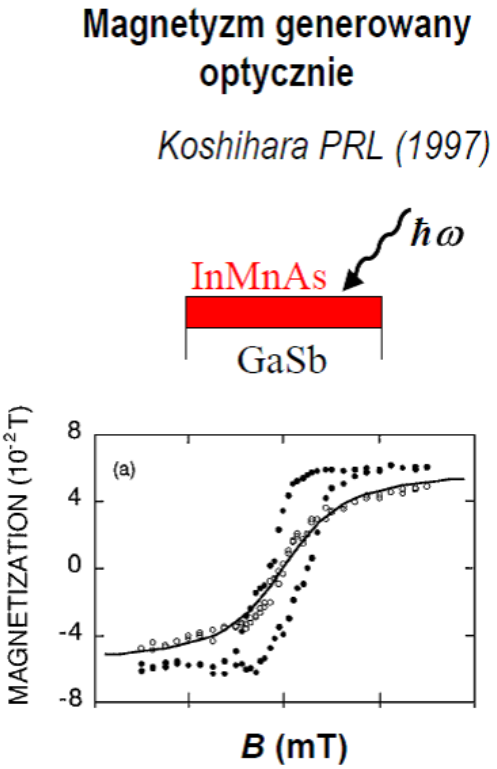
$$\rho(B) = \rho(0)(1 + HB^2)$$

gdzie :

B – indukcja pola magnetycznego

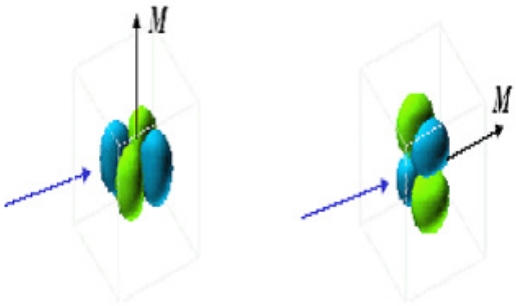
$\rho(B)$ – opór właściwy w polu magnetycznym

$\rho(0)$ – opór właściwy bez pola magnetycznego.



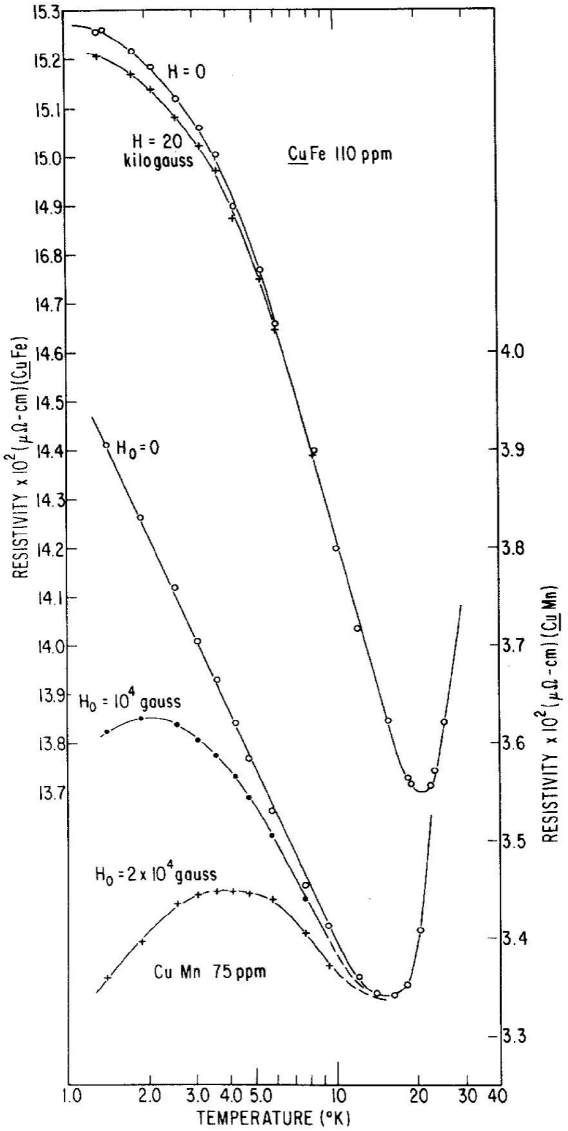
Rysunek 6. Indukowany ferromagnetyzm w InMnAs światłem o energii $E = h\sqrt{}$ [6].

Zależność oporu elektrycznego w zwykłych metalach i półprzewodnikach od pola magnetycznego pojawia się w silnych polach magnetycznych, więc ich zastosowanie w spin elektronice nie jest możliwe. Ponadto w metalach ferromagnetycznych (metalach przejściowych) opór elektryczny zależy od kierunku prądu względem kierunku namagnesowania (anizotropowy magnetoopór). Ten anizotropowy magnetoopór wynika z obecności elektronów 3d. Namagnesowanie w tym przypadku wpływa na orbitale 3d (sprzężenie spin – orbita). Orbitale zmieniają orientację w polu magnetycznym (rys. 7).



Rysunek 7. Zmiana orientacji orbitali 3d w polu magnetycznym B

Wprowadzenie niewielkiej ilości domieszek magnetycznych do stopów ze zwykłymi metalami może spowodować powstanie magneto oporu. Pojawia się efekt Kondo rys. 8



Rysunek 8. Zależnośćć oporu elektrycznego R od temperatury dla stopów Cu domieszkowanych atomami Mn i Fe.

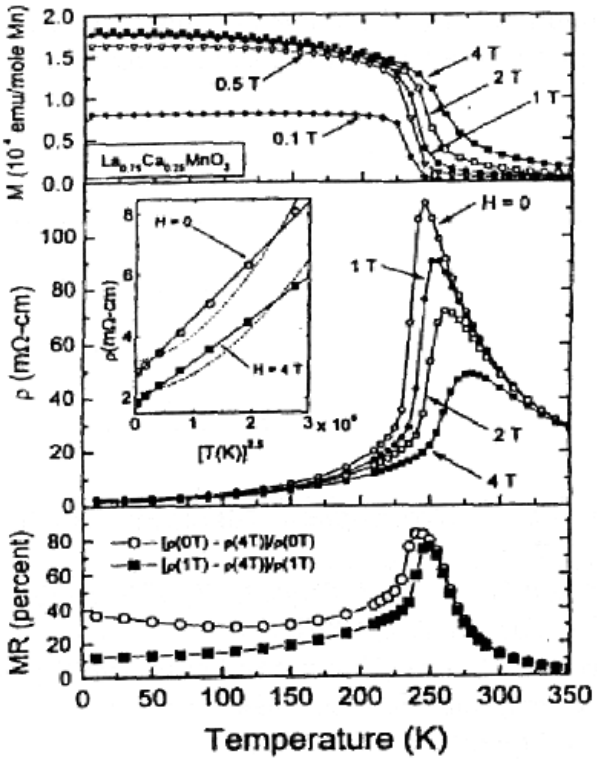
Wprowadzenie do półprzewodnika niewielkiej ilości domieszek magnetycznych powoduje powstanie uporządkowania magnetycznego w półprzewodnikach. Takie półprzewodniki magnetyczne mogą stanowić najważniejsze materiały dla spintroniki.

3. Materiały spintroniczne

Do materiałów mogących mieć zastosowanie w spintronice należy zaliczyć:
1. metale ferromagnetyczne,
2. półprzewodniki magnetyczne (perovskity),
3. półprzewodniki magnetyczne *EuX*,
4. półprzewodniki magnetyczne DMS (diluted magnetic semiconductors),
5. nanostruktury, nanorurki, grafeny.

Metale ferromagnetyczne nie są nowymi materiałami ale w spin elektronice są stosowane ponieważ
- oddziaływanie wymiany powoduje, że koncentracja elektronów o spinie ↑ i spinie ↓ może być różna,
- mają anizotropowy magnetoopór.

Półprzewopdniki magnetyczne o strukturze pervoskitu to manganowe związki $A_{1-x}B_xMnO^3$ gdzie *A = La, Nd, Pr, B = Ca, Ba, Sr.*
Materiały te w pobliżu temperatury Curie wykazują bardzo silny magnetoopór (rys. 9)



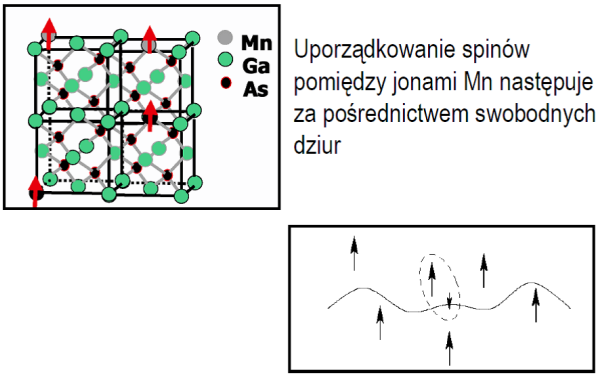
Top:Magnetization against temperature for $La_{0.75}Ca_{0.25}MnO_3$ for various field values
Middle: resisitivity against temperature
Bottom: magnetoresistance against temperature

Rysunek 9. Zależności namagnesowania , oporu właściwego i magnetooporu w funkcji temperatury dla $La_{0.75}Ca_{0.25}Mn0_3$.

Przewodzenie prądu odbywa się w nich przez hooping miedzy jonami Mn^{3+} i Mn^{4+}. Momenty magnetyczne muszą być równoległe aby to było możliwe, to znaczy powinien być stan ferromagnetyczny.
W temperaturze Tc zachodzi przemiana izolator –metal a pole magnetyczne zwiększa uporządkowanie ferromagnetyczne, opór elektryczny maleje.
We wczesnych latach 60-tych badano związki typu *EuX* gdzie *X = 0, S, Te,* w których jon magnetyczny Eu^{2+} zajmował położenia w każdym węźle sieci, oraz inne materiały *GdS, EuSe* i spinele $CuCr_2Se_4$.

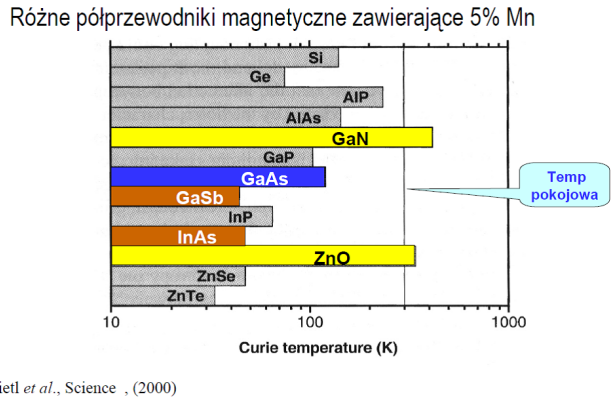
Półprzewodniki magnetyczne tego typu jednak nie znalazły zastosowania w spin elektronice, ponieważ temperatura Curie wynosi około *80K*, trudno jest je syntezować, struktura krystaliczna jest inna niż *Si i GaAs*, małe są nadzieje na poprawę ich własności.

Rozwój półprzewodników DMS (diluted magnetic semiconductors) zaczyna się w latach 80-tych. Są to półprzewodniki w których atomy III grupy Układu Mendelejewa w związkach typu III – V są częściowo zastąpione przez jony magnetyczne np.: Mn, Co. Mogą to być również półprzewodniki II – VI. Uporządkowanie spinów pomiędzy jonami Mn następuje za pośrednictwem swobodnych dziur (rys. 10)



Rysunek 10. Uporządkowanie spinów manganu w MnGaAs.

Zasadniczym problemem jest bardzo trudna synteza takich półprzewodników. Bardzo trudno je domieszkować tak aby otrzymać półprzewodniki magnetyczne typu *n, p* a potem stosować je w elektronice. Inny problem ich zastosowania to niskie temperatury Curie. Na rys. 11 przedstawiono różne półprzewodniki magnetyczne. Z rys. 11 wynika, że nadzieje budzą półprzewodniki GaN i ZnO. Ostatnio potwierdzono, że GaMnN jest ferromagnetykiem z T_c = 300K.



Dietl *et al.*, Science , (2000)

Rysunek 11. Temperatury Curie dla wybranych półprzewodników magnetycznych (7)

Ostatnio są prowadzone intensywne badania zastosowania krzemu w urządzeniach spintronicznych o własnościach magnetycznych. Badania prowadzone przez Vincenta LaBella i Martina Bolduc pokazały, że Si implantowany Mn (koncentracja do 1%) ma własności magnetyczne aż do 127oC.

Niezwykle ciekawą grupę materiałów magnetycznych stanowią niedawno odkryte materiały, które nie zawierają pierwiastków magnetycznych ale wykazują uporządkowanie magnetyczne poniżej 300 K. W tych materiałach oddziaływania wymienne są spowodowane sprzężeniem elektronów sp2 na orbicie ze spinem. Należą do nich (CaLa)B6 polimeryzowany C60 (TDA-E-C60) i ostatnio grafen [3]. W romboedrycznym C60 stwierdzono ferromagnetyczne uporządkowanie z Tc = 500K pod wysokim ciśnieniem [8]. Możliwość wystąpienia magnetycznego uporządkowania w grafenie i nanostrukturach przedstawił Oleg V Yazyev w teoretycznych rozważaniach [3], oraz ich możliwego zastosowania w spintronice. Grafen jest bardzo dobrym przewodnikiem ciepła (4840 – 5300 W/mK), podczas gdy srebro ma 429 W/mK, posiada niewielką rezystancję. Bardzo wysoka ruchliwość elektronów w temperaturze pokojowej przy założeniu, że rozpraszanie jest jedynie na fononach wynosi 8500 cm2/Vs (dla krzemu 1500 cm2/Vs, arsenek galu 8500cm2/Vs). Prędkość przepływu elektronów wynosi 1/300 prędkości światła, umożliwia badanie efektów relatywistycznych dla elektronu poruszającego się w przewodniku. Jest prawie że przezroczysty, pochłania tylko 2% światła. Jest prawie 100 razy mocniejszy niż stal, a zarazem tak elastyczny, że można go rozciągać do 20%. Grafenu w stanie wolnym nie można otrzymać ze względu na jego nietrwałość i skłonność do tworzenia struktur trójwymiarowych (fulerenów i nanorurek). Otrzymuje się go metodami mikromechanicznymi. Grafen do niedawna był najdroższym materiałem na Ziemi. Cena mikromechanicznie odłupanego krystalitu grafenu wielkości powierzchni przekroju włosa ludzkiego kosztowała 1000 dolarów. Koreańczycy z Uniwersytetu Sungkyunkwan opracowali metodę pozwalającą na tańsze wyprodukowanie fragmentów grafenu o powierzchni nawet 1 cm2.

4. Wstrzykiwanie i manipulacja spinem

Wstrzykiwanie spinu może być z metalu do metalu. Najprostszą metodą jest wstrzykiwanie elektronów z ferromagnetyka w którym dominuje pewien kierunek ułożenia spinów do niemagnetycznego półprzewodnika poprzez kontakt elektryczny. Okazuje się, że efektywność tego procesu jest bardzo mała z powodu niedopasowania pasm energetycznych, a co za tym idzie energii nośników oraz ich koncentracji. Problemem jest również relaksacja (uporządkowanie spinów zanika w

czasie). Lepszym rozwiązaniem jest tworzenie złącz ferromagnetycznego metalu i półprzewodnika takich przez które elektrony tunelują. Ferromagnetyczne uporządkowanie możemy wywołać światłem zwłaszcza w przypadku heterostruktur (AlGaMnSb).

5. Urządzenia spintroniczne

Początki rozwoju spintroniki datuje się od wykrycia gigantycznego magnetooporu w metalach. Peter Grunberg (Niemcy) i Albert Fert (Francja) w roku 2007 otrzymali nagrodę nobla w dziedzinie fizyki za wykrycie magnettooporu w układzie trójwarstwowym Fe/Cr/Fe. Okazało się, że im więcej było warstw tym silniejsza była zmiana magnetooporu [9, 10]. Opór układów wielowarstwowych złożonych z magnetyka i niemagnetyka silnie zależy od pola magnetycznego. Przyczyną gigantycznego magnetooporu jest zależność rozpraszania elektronów od spinu. Elektrony o danej orientacji spinu są silnie rozpraszane w warstwie o pewnym kierunku namagnesowania a słabo w warstwie o przeciwnym namagnesowaniu.

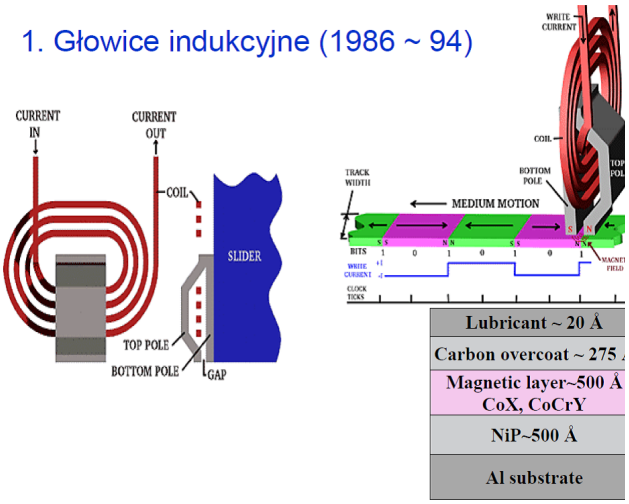
W przypadku sprzężenia ferromagnetycznego warstw elektrony o spinie zorientowanym w dół mogą przepływać przez układ. Mamy do czynienia z niskim oporem układu. W przypadku antyferromagnetycznego sprzężenia warstw elektrony o spinie zorientowanym w dół nie mogą przepływać przez układ i mamy przypadek powstania bardzo dużego magnetooporu. Różnica oporów dla układu wielowarstwowego może sięgać kilkudziesięciu procent. Gigantyczny magnetoopór może powstawać gdy warstwy niemagnetyczne są wąskie (węższe niż droga swobodna elektronu [11, 12]). W przypadku gdy dwie warstwy ferromagnetyczne są oddzielone od siebie warstwą izolatora może wystąpić tunelowanie. Tunelowanie zachodzi zazwyczaj bez zmiany orientacji spinu. Większość elektronów na poziomie Fermiego ferromagnetyka ma jeden kierunek spinu, zatem prąd tunelowy jest spolaryzowany pod tym względem. Opór złącza tunelowego też zależy od tego czy ferromagnetyczne warstwy są namagnesowane zgodnie czy przeciwnie. Jeżeli potrafimy sterować namagnesowaniem to będziemy mieć urządzenia spintroniczne. tzw. zawór spinowy (spin valve).

6. Zastosowanie GMR (gigantycznego magnetooporu)

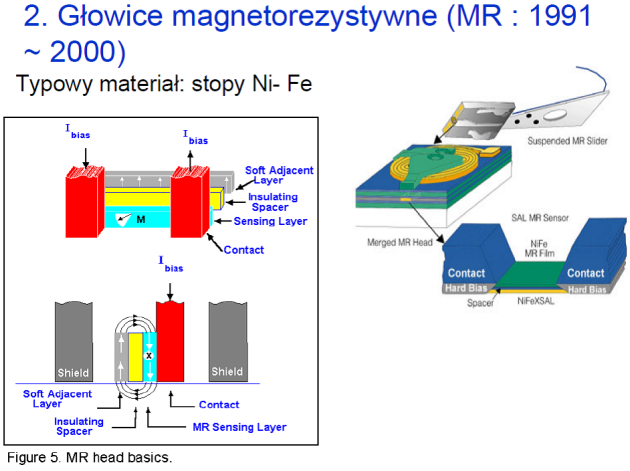
1. Pomiar pola magnetycznego w sterownikach dysków magnetometrach kompasach
2. detekcja położenia – sensor mierzy zmianę pola magnetycznego związana z przemieszczaniem czegoś co wytwarza pole np. magnes na wale silnika spalinowego (obecnie są stosowane sondy hallowskie).
3. Głowice – od głowic indukcyjnych do zaawansowanych głowic GMR.
4. Twarde dyski:

1973 – pierwszy twardy dysk o nowoczesnej konstrukcji model IBM 3340"Winchester o pojemności 60 MB
1983 pierwsza dyskietka 3,5".

Na rys.12 przedstawiono zasadę działania głowicy indukcyjnej, którą stosowano najczęściej w latach 1986 – 1994 w komputerach



Rysunek 12. Schemat i budowa głowicy indukcyjnej stosowanej w komputerach w latach 1986-1994.

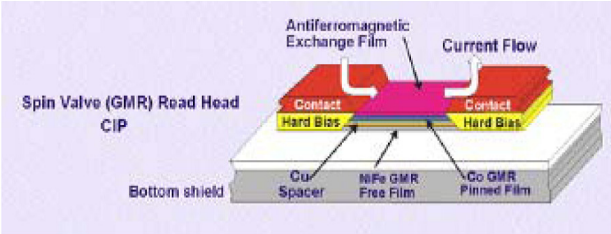


Rysunek 13. Schemat glowicy magnetorezystywnej stosowanych w latach 1991-2000 w komputerach

Głowicę magnetorezystywne najczęściej stosowane w latach 1991-2000 (rys. 13) bazują na anizotropowym magnetooporze, w których $\Delta R/R = 2 – 5\%$ co daje od 1 do 5 Gb/ *sq.inch*. Po raz pierwszy wprowadzone zostały przez IBM. . Materiałem typowym stosowanym w budowie głowicy jest stop *Ni-Fe*.
Od 1997 roku stosuje się głowice z gigantycznym magnetooporem $\Delta R/R = 10 – 50\%$co daje *10Gb/sq.inch*

3. Głowice z gigantycznym magnetooporem (od 1997)

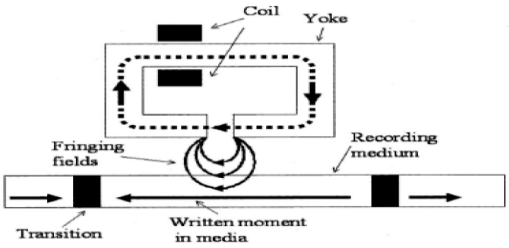
⌘ΔR/R=10~50%, co daje 10Gb/sq.inch



http://www.owlnet.rice.edu/~phys533/notes/week14_lectures.pdf

Rysunek 14. Przekrój warstwowy głowicy z gigantycznym magnetooporem

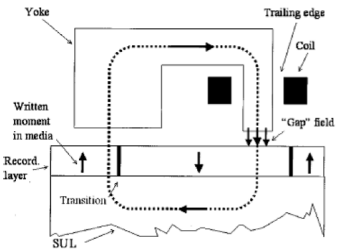
Głowica: zapis podłużny



S. Khizroev and D. Litvinov, J.A.P Vol 95,Num 9, May 2004

Rysunek 15. Schemat działania głowicy przy zapisie podłużnym [13]

Głowica: zapis poprzeczny



S. Khizroev and D. Litvinov, J.A.P Vol 95,Num 9, May 2004

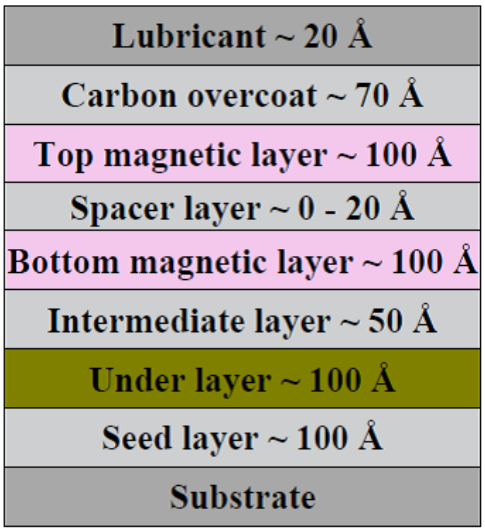
The first one use perpendicular is Toshiba's mini hard drive MK8007GAH , which will be used in IPod, 80GB 1.8in

Rysunek 16. Schemat działania głowicy przy zapisie poprzecznym.[13]

Na rys. 14 przedstawiono schemat takiej głowicy. Kolejne rys. 15, 16 przedstawiają zapisy podłużny i poprzeczny [13].

Materiały magnetyczne do budowy głowic muszą spełniać pewne wymagania, muszą to być magnetyki o małym ziarnie krystalicznym (CoCrPtTa , CoCrPtB). Na rys. 17 pokazano przekrój przez materiał głowicy.

Do budowy twardego dysku stosuje się warstwę magnetyczną stop Co o grubości 10 – 30 nm naniesiony na podłoże Al.-Mg z warstwą NiP lub szkło. Podłoże najpierw jest powlekane cienka warstwą Cr lub stopu CrV w celu zapewnienia odpowiedniej orientacji krystalograficznej warstwy magnetycznej. Gładkość powierzchni twardego dysku powinna wynosić kilka nm, zaś przerwa pomiędzy głowicą a twardym dyskiem wynosić powinna około 15 nm.



Rysunek 17. Warstwowy przekrój głowicy

Pytanie stawiane przez naukowców często sprowadza się do „Czy istnieje granica możliwości pojemności twardego dysku."

Każdy bit zawiera setki ziaren krystalicznych. Zapis magnetyczny polega na uśrednieniu namagnesowania wszystkich ziaren. Gdy bity maleją to ziarna też muszą maleć. W końcu stają się superparamagnetyczne. Superparamagnetyzm polega na tym, że magnetyczna informacja zawarta w ziarnie ulega z pomocą energii termicznej spontanicznemu przełączaniu. Aby zachować informację dłuższą niż 10 lat magnetyczna anizotropia ziarna $K_uV > 40\text{-}50kT$, oznacza to, że gdy V maleje to K_u musi rosnąć.

Magnetyczna anizotropia może być zwiększona poprzez wytwarzanie materiałów o małym rozrzucie wielkości ziarna. Zapis magnetyczny prostopadły pozwala na użycie większego pola zapisu. Przy pomocy lokalnego ogrzewania materiału dysku można obniżyć pole koercji.

Naukowcy wiążą nadzieję z niedawno odkrytym grafenem. Materiał ten może zastąpić krzem. W Massachusetts Institute of Technology (MIT) zbudowano grafenowy układ (mnożnik częstotliwości) pozwalający odebrać sygnał elektryczny pewnej częstotliwości i wyprodukować sygnał będący wielokrotnością tej częstotliwości. Testy wykazały, że tranzystor wytworzony w procesie technologicznym na bazie grafenu 240 nm

jest w stanie osiągnąć częstotliwość do 100 GHZ. Czujniki grafenowe potrafią zarejestrować obecność pojedynczej cząsteczki szkodliwej substancji. Stosując go w urządzeniach spintronicznych możemy otrzymywać materiały magnetyczne z bardzo dużym magnetooporem gdy ma on kontakt z antyferromagnetycznymi warstwami lub mały opór gdy jest wkomponowany pomiędzy ferromagnetyczne warstwy. Son et al. [14] pokazał, że w nanodrutach grafenu polem elektrycznym można wymusić przejście half-metallicity. Ten stan odpowiada współistnieniu stanu metalicznego dla spinów zorientowanych w górę i izolatora dla spinów zorientowanych przeciwnie.

Pole elektryczne przyłożono poprzecznie do nanorudrutu z grafenu. W polu zerowym układ jest scharakteryzowany przez przerwę energetyczna Δs dla spolaryzowanych spinów na końcach nanodrutu grafenu spinów. Przyłożone pole elektryczne narusza symetrie energetyczną i prowadzi do zniszczenia przerwy energetycznej Wartość krytycznego pola elektrycznego odpowiadająca przejściu half-metallicity jest 3.0/w V gdzie w jest przekrojem nanodrutu. To zjawisko może być wykorzystywane do budowy prostego czujnika elektrycznego transportu spinu.

Nowe odkryte materiały magnetyczne w nanostrukturach, nanodrutach, nanorurkach będą miały duże zastosowanie w mikroelektronice. Otrzymywanie tych materiałów na skale przemysłową jest trudne ale możliwe.

Literatura (References)

[1] K. A. Gsschnejder, L. Eyring, *Handbook on the Physics and Chemistry of Rare Earths*. Vol. 12. North-Holland, Amsterdam 1989.

[2] J. Mulak, W. Suski, R. Troć, *2nd International Conference on the Electronic Structure of the Actinides proceedings*, September 13-16, 1976. Ossolineum, Wrocław 1977.

[3] O. V. Yazyev, *Emergence of magnetism in grapheme materials and nanostructures*, Reports on Progress in Physics 73 (2010), 1-15.

[4] K. S. Novoselov, A. K. Geim, S. V. Morozov, D. Jiang, Y. Zang, S. V. Dubonos, I. V. Grigoriewa, A. A. Firsow, *Electric Field Effect in Atomically Thin Carbon Films*, Science 306 (2004), 666-669.

[5] Z. Wilamowski, *Spintronika*. *Postępy Fizyki* tom 55, zeszyt 3 (2004), 115-119.

[6] S. Koshihara, A. Oiwa, M. Hirasawa, S. Katsumoto, Y. Iye, C. Urano, H. Takagi, H. Munekata, *Ferromagnetic Order Induced by Photogenerated Carriers in Magnetic III-V Semiconductor Heterostructures of (In,Mn)As/GaS*,. Phisical Review Letters 78 (1997), 4617-4620.

[7] T. Dietl, H. Ohno, F. Matsukura, J. Cibert, D. Ferrand, *Zener Model Description of Ferromagnetism in Zinc-Blende Magnetic Semiconductors*, Science 287 (2000), 1019-1022.

[8] T. L. Makarova, B. Sundquist, R. Hohne, P. Esquinazi, Y. Kopelevich, P. Scharff, V. A. Davydov, L. S. Kashevarova, A. V. Rakhmanina, *Magnetic carbon,* Nature 413 (2001), 716-718.

[9] P. Grünberg, *Light Scattering from Spinwaves in Layered Magnetic Structures. Light Scattering in Solids V*, (red.) M. Cardona, G. Güntherodt, Applied Physics Letters 66 (1989).

[10] M. N. Baibich, J. M. Broto, A. Fert, F. Nguyen, P. Van DauF, F. Petroff, *Giant Magnetoresistance of (001) Fe/(001) Cr Magnetic Superlattices*, Phisical Review Letters 61, 21 (1988), 2472-2475.

[11] M. Bowen, M. Bibes, A. Barthélémy, J.-P. Contour, A. Anane, Y. Lemaître, A. Fert, *Nearly total spin polarization in La2/3Sr1/3MnO3 from tunneling experiments*, Applied Physics Letters 82 (2001), 233-236.

[12] S. S. P. Parkin, Ch. Kaiser, A. Panchula, P. M. Rice, B. Hughes, M. Samant i S.-H. Yang, *Giant tunnelling magnetoresistance at room temperature with MgO (100) tunnel barriers*, Nature Materials 3 (2004), 862-867.

[13] S. Khizoroev and D. Litwinov, JAP 95,9, (2004).

[14] Y. W. Son, M. L. Cohen, S. G. Louie, *Half-metallic grapheme nanoribbins*. Nature 444 (2006), 347-349.