

Metody numeryczne w elektrotechnice

Ćwiczenie 5 — Aproksymacja

Dariusz Borkowski

18 marca 2014

Wstęp

Na zajęciach rozwiążecie 3 zadania. Pierwsze pokaże wam zalety stosowania baz funkcji innych niż jednomiany w aproksymacji wielomianami wyższych stopni. Drugie pokazuje różnice pomiędzy aproksymacją jednostajną i średniokwadratową na podstawie projektowania filtrów cyfrowych. Trzecie zadanie nauczy was korzystania z biblioteki GSL do aproksymacji wielomianowej na przykładzie rzeczywistej charakterystyki czujników nacisku.

1 Aproksymacja średniokwadratowa punktów równoodległych z bazą wielomianów ortogonalnych – 3 pkt

W praktyce bardzo często spotykamy się z potrzebą wygładzenia (aproksymacji) punktów pomiarowych wyznaczonych z błędami. Aproksymacja średniokwadratowa $N + 1$ punktów o współrzędnych x_i, y_i ma postać

$$F(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + a_2\varphi_2(x) + \dots + a_M\varphi_M(x) \quad (1)$$

gdzie $\varphi_0, \varphi_1, \varphi_2, \dots, \varphi_M$ są funkcjami bazowymi $M+1$ wymiarowej podprzestrzeni liniowej, a współczynniki $a_0, a_1, a_2, \dots, a_M$ są tak dobrane aby minimalizować błąd średniokwadratowy

$$\frac{1}{N+1} \sum_{i=0}^N |F(x_i) - y_i|^2 \quad (2)$$

Funkcjami bazowymi $\varphi_m(x)$ wielomianu uogólnionego $F(x)$ stopnia M mogą być np. jednomiany $1, x, x^2, \dots, x^M$ lub inne funkcje. Duże znaczenie mają bazy złożone z wielomianów ortogonalnych (np. wielomiany Grama) lub funkcji trygonometrycznych.

Aby wyznaczyć funkcję aproksymującą należy wybrać bazę funkcji $\varphi_m(x)$ i obliczyć wartości współczynników $a_0, a_1, a_2, \dots, a_M$. Związek pomiędzy funkcją aproksymującą $F(x)$ a punktami x_i, y_i można zapisać w postaci równania

$$\begin{bmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \dots & \varphi_M(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \dots & \varphi_M(x_1) \\ \varphi_0(x_2) & \varphi_1(x_2) & \dots & \varphi_M(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_0(x_N) & \varphi_1(x_N) & \dots & \varphi_M(x_N) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_M \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \quad (3)$$

lub krócej

$$\mathbf{V}\mathbf{a} = \mathbf{y} \quad (4)$$

gdzie $\mathbf{V}$ nazywamy macierzą Vandermonde'a jeśli $\varphi_m(x) = x^m$ i $M = N$, $\mathbf{a}$ jest wektorem poszukiwanych współczynników, $\mathbf{y}$ jest wektorem wartości aproksymowanych punktów.

Jeżeli $M < N$ (układ równań nadokreślony, typowy dla aproksymacji) to równanie to można rozwiązać w sensie najmniejszych kwadratów (LS) po przemnożeniu go obustronnie przez macierz $\mathbf{V}^T$. Wtedy otrzymamy tzw. równanie normalne

$$\mathbf{V}^T \mathbf{V} \mathbf{a} = \mathbf{V}^T \mathbf{y} \quad (5)$$

którego rozwiązanie ma postać

$$\mathbf{a} = (\mathbf{V}^T \mathbf{V})^{-1} \mathbf{V}^T \mathbf{y} \quad (6)$$

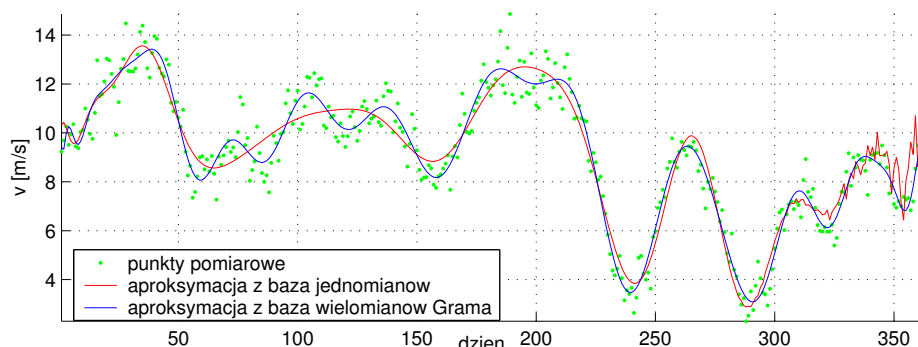
Od wyboru funkcji bazowych zależy postać macierzy $D = V^T V$ równania normalnego (5). W najłatwiejszym i najczęściej stosowanym przypadku przyjęcia bazy złożonej z jednomianów, macierz D dla dużych M ma wyznacznik bliski zeru i jest źle uwarunkowana. Powoduje to, że dla dużych M współczynniki a mogą być obciążone tak dużymi błędami, że efekt aproksymacji będzie do niczego.

Złego uwarunkowania dla wysokiego stopnia M wielomianów aproksymujących można uniknąć przyjmując jako bazę funkcje ortogonalne na punktach x_i (bądź na przedziale w przypadku aproksymacji funkcji). W przypadku idealnie dokładnych obliczeń, pozbawionych zaokrągleń związanych z reprezentacją danych, wyniki aproksymacji z różną bazą są jednakowe. Efektem przyjęcia funkcji ortogonalnych w przedziale aproksymacji jest dobrze uwarunkowana, diagonalna macierz D oraz prawidłowo wyznaczone współczynniki a . Jeśli funkcje bazowe są ortonormalne, czyli ortogonalne unormowane do jedności (normalizację przeprowadzamy dzieląc wartości funkcji przez jej normę l^2 równą $\sqrt{\sum \varphi(x_i)^2}$), wtedy macierz D staje się macierzą jednostkową I , a rozwiązanie (6) upraszcza się do postaci

$$a = V^T y \quad (7)$$

i może być tym samym wyznaczone znacznie szybciej (bez odwracania macierzy lub jej dekompozycji).

W większości przypadków do przybliżenia danych pomiarowych wystarcza aproksymacja wielomianem niskiego stopnia (np. $M < 10$). W niektórych przypadkach wymagana jest aproksymacja wielomianem znacznie wyższego stopnia. Przypadek taki pokazano na rysunku 1. Rysunek przedstawia prędkości wiatru zarejestrowane w ciągu całego roku w punkcie pomiarowym na potrzeby weryfikacji lokalizacji elektrowni wiatrowej. Dokładność pomiarów jest niska (błąd graniczny wynosi 1 m/s) przez co wyniki mogą być traktowane jako sygnał zawierający szum. Z tego względu zdecydowano się na aproksymację zaburzonych danych pomiarowych. Twoim zadaniem będzie porównanie aproksymacji wielomianami wysokiego stopnia wyznaczonym z wykorzystaniem różnych funkcji bazowych.



Rysunek 1: Aproksymacja prędkości wiatru w punkcie pomiarowym w ciągu roku wielomianami 35 stopnia.

Ładuj szkielet programu `lab05z01.m`. Do pracy potrzebne będą też dane zapisane w pliku `lab05z01.mat` oraz funkcje `jedn.m`, `gram.m`, muszą być zapisane w tym samym katalogu co plik `lab05z01.m`. Na wstępie po załadowaniu danych wybierz z obu wektorów (x i y) pierwsze 25 próbek. Korzystając z funkcji `jedn.m` zbuduj macierz V i wyznacz współczynniki wielomianu aproksymującego rozwiązując równanie $Va = y$.

Funkcja `jedn.m` przyjmuje jako argumenty wektor x i stopień wielomianu m , a zwraca wektor wartości jednomianów x^m , który będzie m -tą kolumną macierzy V . Funkcja `gram.m` przyjmuje działa w identyczny sposób, ale wyznacza wektor wartości wielomianów Grama¹ $g_m(x)$ stopnia m .

Dopisz część programu wyznaczającą rozwiązanie (wystarczy wyznaczanie wartości funkcji aproksymującej w punktach x_i wektora x). Dopisz fragment, wyznaczający wartości błędu średniokwadratowego oraz maksymalnego błędu bezwzględnego aproksymacji. Zwiększaj stopień M wielomianu aproksymującego, rysując jednocześnie efekt aproksymacji, od 1 do kiedy do momentu aż zauważysz, że funkcja aproksymująca jest niewłaściwie dopasowana do punktów. Sprawdź również od jakiego stopnia M wielomianu Matlab ostrzega o złym uwarunkowaniu macierzy D . Przyjrzyj się wartościom elementów macierzy D i policz jej wyznacznik. Następnie zamień funkcję `jedn.m` na `gram.m` i wykonaj te same kroki co poprzednio. Czy Matlab zgłasza złe uwarunkowanie macierzy D ? Czy przy tym samym stopniu M , dla którego wcześniej wyniki aproksymacji były niepoprawne, teraz również są niepoprawne? Jako punkt odniesienia wyznaczaj drugi wielomian aproksymujący funkcję `polyfit`, która zresztą wykorzystuje bazę jednomianów.

¹Wielomiany Grama to wielomiany ortogonalne na zbiorze punktów równoodległych opisane na stronie 86 w [1]

Zwróć uwagę na rozbieżności wyników aproksymacji z bazą jednomianów (funkcja jedn.m) i aproksymacją z tą samą bazą za pomocą polecenia polyfit dla $M \geq 15$. W celu wyjaśnienia przyczyny tej rozbieżności otwórz źródło polecenia polyfit (edit polyfit) i zobacz jaka operacja jest wykonywana na węzłach przez aproksymację (podpowiedź znajdziesz też w instrukcji do zadania lab04z01).

Zauważyłeś zapewne, że aproksymacja z wielomianami Grama daje lepsze efekty dla dużych M , ale pociąga za sobą znacznie dłuższy czas obliczeń. Można ten czas skrócić stosując ogólną zależność iteracyjną² (tzw. metodę trójczołową), która wiąże ze sobą wielomiany ortogonalne różnych stopni. Dla wielomianów Grama wyprowadzono gotowe zależności rekurencyjne, z których otrzymujemy wielomiany ortonormalne (czyli znormalizowane wielomiany ortogonalne):

$$\begin{aligned}\varphi_{-1}(t) &= 0, & \varphi_0(t) &= \frac{1}{\sqrt{N+1}}, \\ \varphi_m(t) &= \alpha_m t \varphi_{m-1}(t) - \gamma_m \varphi_{m-2}(t),\end{aligned}\quad (8)$$

gdzie t_i są punktami z przedziału -1 do 1 (na tym przedziale wielomiany Grama są ortogonalne). Dysponując $N+1$ węzłami x_i ($i = 0 \dots N$) obliczamy wartości punktów t_i jako $2\frac{i}{N} - 1$. Współczynniki występujące w równaniach (8) wyznacza się następująco:

$$\alpha_m = \frac{N}{m} \sqrt{\frac{4m^2 - 1}{(N+1)^2 - m^2}}, \quad \gamma_m = \frac{\alpha_m}{\alpha_{m-1}}, \quad (9)$$

Zmodyfikuj program tak, aby tworzyć macierz V w oparciu w.w. zależności rekurencyjne (8–9). Sprawdź czy otrzymane wielomiany są ortonormalne, a następnie wykorzystaj w aproksymacji uproszczony wzór (8).

Porównaj czasy obliczeń dla wielomianu aproksymującego 22 stopnia z definicyjnym i rekurencyjnym wypełnianiem macierzy V . Wykonaj aproksymację (szybszą metodą!) wielomianem stopnia 75 i wyrysuj wyniki. Wyznacz błędy maksymalny (10) i skuteczny czyli pierwiastek z błędu średniokwadratowego (11) tej aproksymacji.

Zauważ, że aproksymacja wysokiego rzędu ($M > 35$) wielomianami Grama obliczanymi z definicji daje gorsze wyniki (błędy) niż przy rekurencyjnym generowaniu wielomianów Grama. Odpowiedz na pytanie znajdziesz po przeanalizowaniu pliku gram.m wyznaczającego wielomiany Grama wg definicji wykorzystującą dwumian Newtona i operację silnia (przeczytaj help do polecenia factorial).

2 Zastosowania aproksymacji średniokwadratowej i jednostajnej – 2 pkt

Zadaniem aproksymacji jednostajnej jest takie dobranie funkcji aproksymującej, która będzie minimalizować maksymalną odchyłkę funkcji aproksymującej od punktów aproksymowanych (bądź funkcji aproksymowanej). Kryterium jakości aproksymacji jednostajnej, zwane też kryterium min-max, ma postać błędu maksymalnego (równe jest normie l_∞ z wektora odchyłek):

$$\|F(x_i) - y_i\|_\infty = \max_i (|F(x_i) - y_i|) \quad (10)$$

Potrzeba stosowania aproksymacji jednostajnej występuje rzadziej niż LS i jest związana z bardzo ostrymi wymaganiami względem funkcji aproksymowanej. Przykładem może być aproksymacja zdjętej charakterystyki, której wartości zostały zmierzone z bardzo dużą dokładnością, mająca na celu zbudowanie modelu matematycznego odwzorowującego dane zjawisko w całym przedziale aproksymacji z dokładnością nie mniejszą niż zadana wartość (czyli najczęściej błąd aproksymacji jednostajnej). Innym zastosowaniem aproksymacji jednostajnej jest aproksymacja skomplikowanych funkcji oraz projektowanie filtrów cyfrowych FIR metodą Parks-McLellana z algorytmem Remez.

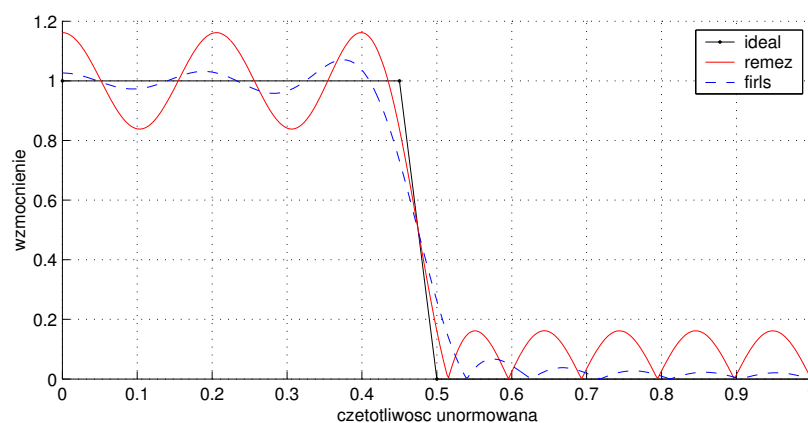
Błąd średniokwadratowy³ (MSE — mean squared error) minimalizowany w aproksymacji średniokwadratowej jest średnią z kwadratów odchyłek i ma postać normy l_2 z wektora odchyłek o długości N :

$$\|F(x_i) - y_i\| = \frac{1}{N} \sum_{i=1}^N |F(x_i) - y_i|^2 \quad (11)$$

²Ta zależność iteracyjna sprawdza się również dla punktów nierównoodległych, pod warunkiem, wielomian jest na tych punktach ortogonalny. Więcej na ten temat na na stronie 88 w [1], a o metodzie trójczołowej na stronach 78–81 w [2]

³Istnieje też błąd skuteczny (RMSE czyli root mean squared error) będący pierwiastkiem z błędu średniokwadratowego.

Twoim zadaniem będzie porównanie błędów projektowania dwóch dolnoprzepustowych filtrów FIR tego samego rzędu według różnych kryteriów. Do projektowania filtrów z zastosowaniem aproksymacji średniokwadratowej służy w Matlabie funkcja `firls`, a do projektowania filtrów z zastosowaniem kryterium min-max służy funkcja `firpm` (dawniej polecenie `remez`) wykorzystująca metodę Parka-McClellana do aproksymacji zadanej charakterystyki wielomianami Czebyszewa. Efekt projektowania filtra dolnoprzepustowego tymi metodami pokazano na rysunku 2. Funkcja `firls` minimalizuje sumę kwadratów odchylek wyznaczonej charakterystyki filtra od zadanej idealnej charakterystyki, a funkcja `firpm` minimalizuje maksymalną odchylek wyznaczonej charakterystyki od zadanej idealnej charakterystyki. Proszę zwrócić uwagę, że w wyniku zastosowania do aproksymacji wielomianów Czebyszewa, przecięcia (pierwiastki) wielomianu aproksymującego z charakterystyką zadaną nie są równoodległe, za to odległość ekstremów wyznaczonej charakterystyki od zadanej jest we wszystkich punktach jednakowa, przez co filtry te nazywane są równofalowymi (ang. *equiripple*).



Rysunek 2: Charakterystyki amplitudowo-częstotliwościowe filtrów dolnoprzepustowych 21 rzędu wyznaczonych za pomocą funkcji `firls` i `remez`.

Otwórz szkielet programu z pliku `lab05z02.m`. Uzupełnij program tak aby na podstawie podanego projektu idealnej charakterystyki amplitudowo-częstotliwościowej i rzędu filtra wyznaczyć współczynniki filtrów FIR dwoma w.w. metodami. Wyrysuj (funkcją `freqz`) charakterystyki amplitudowo-częstotliwościowe otrzymanych filtrów i policz ich błędy maksymalny (10) i średniokwadratowy (11). Dla filtra uzyskanego funkcją `firpm` błąd maksymalny będzie mniejszy, a błąd średniokwadratowy będzie większy niż dla filtra uzyskanego z `firls`. Jeśli uzyskałeś inny wynik, to znaczy, że uwzględniłeś za mało punktów charakterystyki (zwracanych przy okazji przez funkcję `freqz`) przy wyznaczaniu błędów. Pamiętaj, że funkcja `firls`, w procesie aproksymacji (optymalizacji) stosuje całkowe, a nie punktowe kryterium jakości (całka jest oczywiście wyznaczana numerycznie). Funkcja `firpm` znajduje minimalizowaną maksymalną odchylek od charakterystyki idealnej metodą przeszukiwania siatki (ang. *grid search*). Tak jak w przypadku numerycznego wyznaczania całki, przeszukiwanie siatki daje tylko przybliżenie wyniku dokładnego.

Oblicz wartość wzmocnienia i przesunięcia fazowego (w stopniach) wprowadzane przez oba filtry dla podstawowej harmonicznej (50 Hz) napięcia próbkowanego z częstotliwością 6400 Hz.

3 Aproksymacja charakterystyki czujników nacisku w C – 2 pkt

W celu budowy systemu mierzącego masę pojazdów w ruchu z kompensacją temperatury została zdjęta charakterystyka czułości czujników nacisku w funkcji temperatury. Otrzymane wyniki pomiarów zapisano w pliku `lab05z03.dat`.

Napisz w C program wyznaczający współczynniki wielomianu aproksymującego pierwsze N punktów charakterystyki wielomianem stopnia M i zapisujący wyznaczone współczynniki wielomianu do pliku `out.dat`. Współczynniki należy zapisać w formacie ASCII w postaci wektora kolumnowego (który da się odczytać w Matlabie) poczynając od współczynnika przy najwyższej potęgze. Dane te następnie załaduj do Matlab'a i wyrysuj wielomian aproksymujący na tle punktów pomiarowych, obszar wykresu (ustawiany funkcją `axis`) ma być trochę większy niż obszar zajmowany przez aproksymowane dane.

Szkielet programu znajduje się w pliku `lab05z03.c`. Poniżej podaję opis niezbędnych funkcji GSL-a związanych z aproksymacją. Program będzie wykorzystywał niektóre typy danych, które poznacie na

wcześniejszych zajęciach (np. macierze i wektory), których oczywiście już nie opisuję, bo zrobiłem to w poprzednich instrukcjach.

`gsl_multifit_linear_workspace * gsl_multifit_linear_alloc (size_t n, size_t p)` ta funkcja zwraca wskaźnik na zaalokowaną pamięć potrzebną do rozwiązania liniowego problemu najmniejszych kwadratów. Pamięć ta jest niezbędna, gdyż rozwiązanie nadokreślonego układu równań liniowych jest tu wykonywane nie jak w zadaniu 1 przez $\mathbf{a} = (\mathbf{V}^T \mathbf{V})^{-1} \mathbf{V}^T \mathbf{y}$ lecz przez tzw. pseudoinwersję macierzy $\mathbf{V}$ z dekompozycją SVD. n określa ilość aproksymowanych punktów, a p ilość parametrów wyznaczanej funkcji aproksymującej.

`int gsl_multifit_linear (const gsl_matrix * X, const gsl_vector * y, gsl_vector * c, gsl_matrix * cov, double * chisq, gsl_multifit_linear_workspace * work)` ta funkcja wyznacza wektor parametrów $\mathbf{c}$ funkcji najlepiej przybliżającej, według kryterium średniokwadratowego (LS ang. *least squares*) model $\mathbf{y} = \mathbf{X}\mathbf{c}$, którego macierz $\mathbf{X}$ i wektor $\mathbf{y}$ zostały zbudowane na bazie danych wyników pomiarowych. Macierz kowariancji wyliczonych parametrów zostaje zapisana w `cov`, a błąd średniokwadratowy (ang. *mean square error*) zwracany do `chisq`. `work` jest alokowanym, opisaną wcześniej instrukcją, obszarem pamięci potrzebnej do obliczeń.

Funkcja ta pozwala wykonywać aproksymację z wykorzystaniem różnych funkcji bazowych (niekoniecznie jednominów, które stosujemy w tym zadaniu). Inne funkcje bazowe uzyskujemy przez odpowiednią konstrukcję macierzy $\mathbf{V}$. Zamiast wpisywać wartości x^m do m -tej kolumny tej macierzy, możesz wpisywać tam wartości innych funkcji zmiennej x .

W GSLu istnieje też funkcja `int gsl_multifit_wlinear` służąca do aproksymacji z użyciem funkcji wagowej i przyjmuje jako dodatkowy parametr wartości funkcji wagowej $w(x_i)$ w aproksymowanych punktach. Jako wartości funkcji wagowej często stosuje się estymowaną wariancję wyników pomiarowych, jeśli nią dysponujemy.

`void gsl_multifit_linear_free (gsl_multifit_linear_workspace * work)` zwalnia pamięć wykorzystaną do aproksymacji. błędów.

4 Zastosowanie FFT do analizy widmowej sygnału okresowego napięcia – 3 pkt

Problemem jest wyznaczenie harmonicznym i interharmonicznym sygnału napięcia, o znanej częstotliwości f_1 (została ona wyznaczona inną metodą), co do którego możemy założyć, że jest on okresowy.

Odpowiednim narzędziem do analizy częstotliwościowej sygnałów okresowych jest DFT czyli aproksymacja z bazą Fouriera czyli ortogonalną bazą funkcji trygonometrycznych lub jego szybsza odmiana FFT. Aby za pomocą skończonego szeregu Fouriera bezbłędnie przybliżyć sygnał okresowy $x(t)$ muszą być spełnione następujące warunki:

1. pasmo częstotliwościowe sygnału $x(t)$ jest ograniczone do częstotliwości nie większej niż częstotliwość ostatniego elementu bazy
2. częstotliwość pierwszego elementu szeregu Fouriera jest równa częstotliwości podstawowej sygnału $x(t)$
3. długość analizy (przedział całkowania, patrz wykład) jest całkowitą wielokrotnością długości okresu T sygnału

W razie niespełnienia któregoś z warunków funkcja aproksymująca, będąca kombinacją elementów szeregu Fouriera, będzie przybliżała sygnał $x(t)$ najlepiej według przyjętego kryterium (np. LS).

W pomiarach cyfrowych sygnał okresowy $x(t)$ o częstotliwości podstawowej f_1 jest próbkowany z częstotliwością f_s i zapisany w postaci dyskretnej $x(n)$.

Spełnienie warunku 1 zapewniamy przez analogową filtrację antyaliasingową, która ogranicza górną częstotliwość sygnału trafiającego do przetwornika A/C do wartości nie większej niż połowa częstotliwości próbkowania, zgodnie z twierdzeniem o próbkowaniu Shannona–Kotelnikowa (zapobiega to zjawisku nakładaniu się widm).

Drugi z warunków jest trudniejszy do spełnienia, gdyż wymaga synchronicznego próbkowania, t.j. takiego gdzie $f_s = M f_1$, gdzie M całkowite. Konieczne jest wyznaczenie częstotliwości f_1 sygnału i możliwości

dostrojenia częstotliwości próbkowania f_s . Spełnienie tego warunku powoduje, że na pełen okres T_1 sygnału $x(n)$ przypada dokładnie M próbek.

Ostatni warunek jest możliwy do spełnienia przy spełnieniu warunku 2. Jeżeli sygnał jest próbkowany synchronicznie (warunek 2), wtedy długość przedziału analizy N (nazywana również długością okna) musi być równa całkowitej wielokrotności okresu t.j. $N = RM$, gdzie R całkowite. W szczególności jeśli $N = M$ wtedy analizie DFT poddajemy tylko jeden okres sygnału.

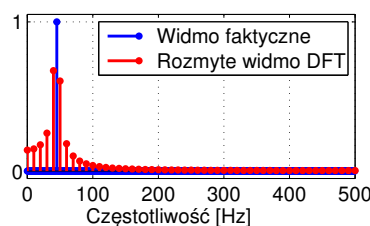
Ze względu na budowę algorytmu FFT (szybkiej implementacji DFT) N powinno być potęgą liczby 2, np. $2^{10} = 1024$, wtedy obliczenia będą najszybsze. Matlabowskie polecenie `fft` radzi sobie jednak z dowolnymi długościami N , wybierając najszybszy dostępny sposób obliczenia DFT.

Wykonując FFT na N próbkach otrzymujemy zawsze N prążków widma (liczb zespolonych zawierających informację o amplitudzie i fazie składowych) rozłożonych równomiernie w przedziale od 0 do $f_s - \Delta f$. Zatem odległości między kolejnymi prążkami określone są wzorem:

$$\Delta f = \frac{f_s}{N}$$

Można też policzyć, że jeśli próbkowanie było synchroniczne $f_s = Mf_1$ to, a długość analizy FFT $N = M$, to wtedy $\Delta f = f_s/N = f_s/M = f_1$ oznacza, że drugi prążek widma odpowiada częstotliwości podstawowej sygnału (pierwszy prążek odpowiada częstotliwości $f = 0$ czyli składowej stałej). Zwiększanie długości N analizy FFT przy niezmięnionej częstotliwości próbkowania powoduje poprawę rozdzielczości częstotliwościowej poprzez zmniejszenie odstępów między prążkami widma (np. $N = 2M$ daje $\Delta f = f_1/2$).

Jeśli próbkowanie nie było synchroniczne lub jeśli było synchroniczne, lecz długość N nie była całkowitą wielokrotnością M , wtedy wystąpi błąd aproksymacji obserwowany w dziedzinie częstotliwości obserwowany jest jako rozmycie widma (rys. 3).



Rysunek 3: Widmo rozmyte na tle widma faktycznego.

W celu ograniczenia rozmycia widma stosujemy tzw. okna czasowe, przez które wymnażamy sygnał przed analizą FFT:

$$x_w(n) = x(n)w(n) \quad (12)$$

Okno czasowe $w(n)$ jest funkcją czasu przyjmującą wartości niezerowe w przedziale analizy T i zero poza nim. Podstawowe okno czasowe prostokątne:

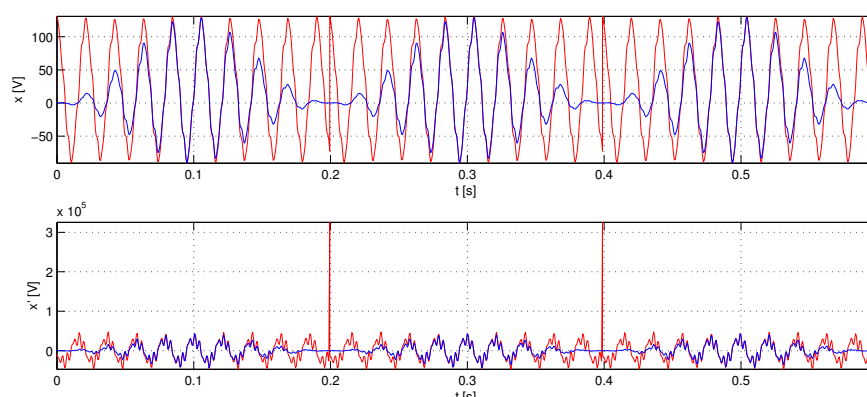
$$w(n) = \begin{cases} 1 & 0 \leq n < N \\ 0 & \text{w pozostałych przypadkach} \end{cases} \quad (13)$$

jest stosowane w zasadzie zawsze w celu wybrania z całego sygnału dokładnie N próbek, które poddajemy FFT. Pozostałe okna czasowe mają postać krzywych, o wartościach malejących do zera na brzegach przedziału N próbek, np. okno Hanninga

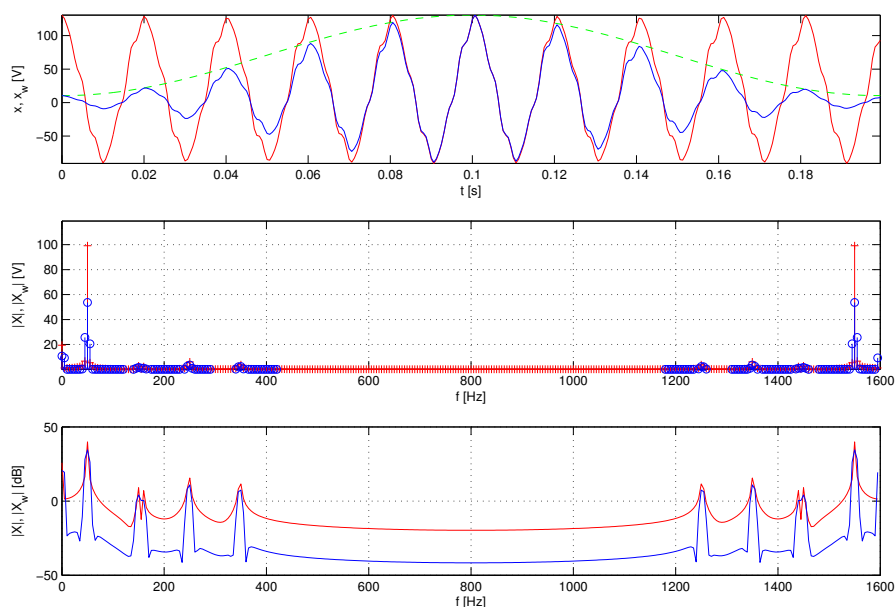
$$w(n) = \frac{1}{2} \cos\left(2\pi \frac{n}{N}\right) \quad 0 \leq n \leq N \quad (14)$$

Wymnożenie sygnału przez okno nieprostokątne ma na celu łagodne doprowadzenie wartości sygnału do zera, przez co nawet jeżeli w analizowanych N próbkach nie mamy pełnego okresu sygnału $x(n)$, to sygnał z nałożonym oknem $w(n)x(n)$ staje się prawie okresowy o okresie N próbek (rys. 4). Dzięki temu rozmycie widma zostaje mocno ograniczone (rys. 5).

Otwórz w Matlabie szkielec programu `lab05z04.m`. Na początek przyjmij, że częstotliwość podstawowa f_1 badanego sygnału $x(t)$ jest równa częstotliwości znamionowej $f_{1N} = 50$ Hz. Za częstotliwość próbkowania f_s przyjmij jej wartość znamionową $f_{sN} = 3200$ Hz, co spełnia założenie próbkowania synchronicznego i daje całkowitą liczbę $M = 64$ próbek na okres sygnału.



Rysunek 4: Przypadek próbkowania niesynchronicznego. Długość N analizy FFT wynosi około 9,5 okresu T sygnału $x(t)$. Sygnał $x(t)$ (czerwony) powielono, w celu pokazania nieciągłości na łączeniach. Sygnał $x_w(t)$ (niebieski) powielono po nałożeniu okna Hanninga. Sygnały zawierają składową stałą. Wykres środkowy to pierwsza pochodna sygnałów, stosowana do wykrywania nieciągłości sygnału.



Rysunek 5: Sygnał $x(t)$ spróbkowany niesynchronicznie z nałożonym oknem prostokątnym (czerwony) oraz z oknem Hanninga (niebieski). Środkowy wykres przedstawia widmo obu sygnałów skali liniowej w V, ostatni w decybelach.

Dobierz najmniejszą długość N analizy FFT, dla której w widmie $|X(f)|$ sygnału $x(t)$ z oknem prostokątnym (kolor czerwony na wykresach Fig. 1) da się rozdzielić prążek związany z harmoniczną rzędu 3 (trzecia harmoniczna) i interharmoniczną rzędu 3,2, a widmo tych składowych nie było rozmyte. Aby widmo nie było rozmyte musisz zapewnić $N = RM$ gdzie R jest całkowite (R jest liczbą analizowanych pełnych okresów sygnału).

Jeśli dobrałeś N prawidłowo prążki widma $|X(f)|$ sygnału $x(t)$ z oknem prostokątnym (czerwone) są rozróżnialne znakomicie, a tłumienie nieistniejących składowych jest na poziomie ok -300 dB. Nieprawidłowy wybór długości N okna analizy FFT będzie widoczny w postaci dużych wartości pochodnych powtórzonego fragmentu sygnału w oknie Fig. 2 w punktach łączenia sygnału.

Porównaj widmo $|X(f)|$ z widmem $|X_w(f)|$ obliczonym dla sygnału $x_w(t)$ z nałożonym oknem czasowym nieprostokątnym (np. Hanninga) (kolor niebieski na wykresach Fig. 1). Zauważ, że pomimo próbkowania synchronicznego i długości N będącej całkowitą wielokrotnością długości okresu, prążki związane z harmonicznymi rzędów 3 i 3,2 „skleiły się” i są nierozróżnialne.

„Sklejenie” prążków widma wynika z własności okien nieprostokątnych, które mają listki boczne niższe niż okno prostokątne, jednak kosztem szerszego listka głównego (w celu wyjaśnienia tych terminów odsyłam do literatury związanej z przetwarzaniem sygnałów i filtracją). Szerszy listek główny okna powoduje, że prążki związane z konkretną składową sygnału również stają się nieco szersze⁴ niż przy oknie prostokątnym. Jest to wada okien nieprostokątnych. Zaletą jest, iż niższe listki boczne okien nieprostokątnych powodują znaczne zmniejszenie rozmycia pomiędzy składowymi nie znajdującymi się zbyt blisko siebie na osi częstotliwości, w sytuacji gdy próbkowanie nie jest synchroniczne.

Zmień częstotliwość podstawową sygnału na $f_1 = 49,9$ Hz, która to wartość jest dosyć prawdopodobna w krajowym systemie energetycznym. Porównaj wynik z uzyskanym poprzednio dla próbkowania synchronicznego. Tym razem lepszy wynik (lepszy w sensie tłumienia nieistniejących składowych) otrzymano dla sygnału $x_w(t)$ na który nałożono okno nieprostokątne (tutaj okno Hanninga). Jednak selektywność (możliwość rozróżnienia składowych) jest nadal lepsza dla okna prostokątnego.

Dobierz częstotliwość próbkowania f_s sygnału tak by spełnione było założenie próbkowania synchronicznego. Jeżeli dla częstotliwości znamionowej $f_1 N$ sygnału znamionowa częstotliwość próbkowania f_{sN} zapewniała synchroniczne próbkowanie, to aby utrzymać synchronizm próbkowania po zmianie częstotliwości f_1 sygnału, zmiana częstotliwości próbkowania f_s powinna być proporcjonalna do odchyłki faktycznej częstotliwości f_1 sygnału od jego częstotliwości znamionowej f_{1N} .

Podsumowując, stosowanie okien nieprostokątnych poprawia możliwość detekcji składowych o małych amplitudach, a czasem także ich rozróżnialność, o ile nie leżą zbyt blisko siebie na osi częstotliwości. Okno prostokątne daje najlepszą rozróżnialność bliskich sobie składowych, ale powinno być stosowane tylko przy synchronicznym próbkowaniu.

Wyrysuj w nowym wspólnym oknie wykresu charakterystykę fazowo-częstotliwościową sygnałów $x(t)$ i $x_w(t)$ (polecenie `angle`) wyrażoną w stopniach (a nie w radianach). Zaznacz za pomocą dodatkowych punktów na wykresach wartości składowych obecnych w sygnale t.j. harmonicznym rzędów 1; 3; 3,2; 5 i 7. Sprawdź czy otrzymane wartości argumentów prążków widma zgadzają się z wartościami kątów fazowych badanego sygnału (generowanego przez plik `source.m`). Zastanów się dlaczego charakterystyka sygnału $x(t)$ jest tak poszarpana, a $x_w(t)$ nie jest. Odpowiedź: jaki jest kąt fazowy wektora o zerowej lub bardzo małej długości? Wróć do próbkowania niesynchronicznego i wyrysuj ch-ki ponownie. Jak wyjaśnisz zaobserwowaną zmianę?

Na następne zajęcia

Całkowanie i różniczkowanie numeryczne.

Literatura

- [1] Fortuna Z., Macukow B., Wąsowski J. *Metody numeryczne WNT*, Warszawa 1993.
- [2] Guziak T., Kamińska A., Pańczyk B., Sikora J. *Metody numeryczne w elektrotechnice* Wydawnictwo Politechniki Lubelskiej, Lublin 2002.
- [3] Bendat J.S., Piersol A.G. *Metody analizy i pomiaru sygnałów losowych*
- [4] Lyons R. *Wprowadzenie do cyfrowego przetwarzania sygnałów*

⁴Dla przykładu przy próbkowaniu synchronicznym i $N = RM$, gdzie R całkowite, dla okna prostokątnego prążek związany ze składową ma wysokość 100, a dla okna Hanninga będzie to prążek o wysokości 50 w otoczeniu dwóch prążków o wysokości 25, razem dają również 100.