

WOJSKOWA AKADEMIA TECHNICZNA
im. Jarosława Dąbrowskiego
WYDZIAŁ MECHATRONIKI I LOTNICTWA



PRACA DYPLOMOWA
STUDIA WYŻSZE

.....
plut. pchor. Dawid BREJECKI
(stopień, imiona i nazwisko studenta)

***Program komputerowy do wyznaczania parametrów fali N
na podstawie zarejestrowanych danych ze strzelań***

***Computer program for determining N-wave parameters
based on recorded data from shootings***

.....
(temat pracy dyplomowej w języku polskim i języku angielskim)

.....
Mechatronika, Eksploatacja przeciwlotniczych zestawów rakietowych
(kierunek i specjalność studiów)

mjr dr inż. Dariusz RODZIK

.....
(stopień wojskowy, tytuł/stopień naukowy, imię i nazwisko promotora pracy dyplomowej)

Warszawa – 2019

(strona celowo zostawiona pusta)

Zadanie do pracy str.1.

Zadanie do pracy str.1.

Spis treści

Wstęp	7
1. Zjawisko powstawania fali N i opis jej parametrów	9
1.1. Powstawanie i charakterystyka fali N	9
1.2. Model matematyczny fali N	12
1.3. Charakterystyki częstotliwościowe fali N	14
1.4. Pomiar fali N	15
1.5. Wpływ warunków środowiska na pomiary parametrów fali N	19
2. Projekt i implementacja algorytmu do wyznaczania parametrów czasowo częstotliwościowych fali N	22
2.1. Ogólne informacje o programie	22
2.2. Główne funkcjonalności programu	22
2.3. Opis działania programu	23
2.4. Podsumowanie rozdziału	33
3. Testy walidacyjne programu	35
3.1. Wstęp do rozdziału	35
3.2. Sprawdzenie poprawności wczytania pliku	35
3.3. Sprawdzenie poprawności obliczeń	36
3.4. Sprawdzenie poprawności generowania wykresów	39
3.5. Sprawdzenie poprawności zapisu oszacowanych parametrów do pliku	41
Podsumowanie	42
Bibliografia	43
Załączniki	44

(strona celowo zostawiona pusta)

Wstęp

W drugiej połowie XX wieku nastąpił gwałtowny wzrost zainteresowania naukowców badaniem fal akustycznych. Te rozchodzące się w ośrodku zaburzenia, powstałe w wyniku wahań gęstości i ciśnienia, wykorzystywane są w coraz to nowszych zastosowaniach, np. kosmetologii. Szczególnie wdzięcznym polem do badań i wdrożeń są zastosowania wojskowe. Mimo niepohamowanego postępu technicznego, w dalszym ciągu współczesne konflikty militarne opierają się głównie na użyciu broni konwencjonalnej. Broń konwencjonalna nieuchronnie emituje fale dźwiękowe, które mogą być wykorzystane przez inteligentnego przeciwnika. Dysponując odpowiednią technologią, na podstawie lokalizacji odgłosów strzałów można określić położenie źródła strzału, prędkość i trajektorię lotu pocisku, a także typ broni palnej i amunicji. Możliwe jest przez to uzyskanie przewagi na polu walki, nie narażając sił i środków na rekonesans. Systemy rejestracji i obróbki dźwięku należą do stosunkowo tanich. Wyniki analizy akustycznej można także wykorzystać na użytek wojsk własnych, na przykład do oceny stopnia wyszkolenia żołnierzy i zastępowania starych metod oceny skuteczności prowadzenia ognia.

Obecnie Zespół Radioelektroniki Wojskowej Akademii Technicznej jest jednym z ośrodków w Polsce, w którym bada się odgłosy wystrzałów i opracowuje metody rejestracji i obróbki powstających fal akustycznych. Rezultatem przeprowadzanych badań eksperymentalnych jest uzyskiwanie dużej ilości plików z danymi, które generują używane systemy pomiarowe. Z pozyskanych w ramach prowadzonych badań danych pomiarowych, w procesie dalszej obróbki i analizy należy wyodrębnić istotne informacje, które będą później wykorzystane do tworzenia nowych algorytmów i modeli matematycznych analizowanych zjawisk. Stąd celem mojej pracy inżynierskiej jest opracowanie programu komputerowego do szacowania parametrów fali akustycznej na podstawie zarejestrowanych w postaci plików danych pomiarowych ze strzelań. Osiągnięcie tego celu może w istotny sposób usprawnić badania prowadzone przez Zespół Radioelektroniki w przyszłości. Temat ten wybrałem z racji mojego zainteresowania analizą danych i możliwością wykorzystania jej w zastosowaniach militarnych. Cel pracy zrealizowałem poprzez opracowanie programu komputerowego w języku *Python*. Program jest funkcjonalnym narzędziem umożliwiającym szybką analizę uzyskanych w wyniku

pomiarów i pozyskanie informacji o parametrach fali N. Wybrałem język programowania *Python* ze względu na jego ogólną dostępność, funkcjonalność w analizie danych, światowe trendy w jego stosowaniu oraz prostą i spójną składnię. Szczegółowe cele działania opracowanego programu oraz opis zaimplementowanego algorytmu zostały przedstawione w rozdziale drugim. Do opracowania algorytmów programowych wymagane było zrozumienie zachowywania się charakterystycznych fal akustycznych generowanych przez poruszający się z prędkością naddźwiękową pocisk, ich dyfrakcji, absorpcji, odbicia od ziemi i innych powierzchni. Teorię odnośnie powstawania szczególnego rodzaju fal akustycznych (tzw. fal N) przedstawiłem w rozdziale pierwszym. Z uwagi na niewystarczającą ilość literatury polskiej w zakresie rozpatrywanego zjawiska, wiedzę swoją opierałem głównie na artykułach amerykańskich naukowców. Ostatni rozdział pracy zawiera opis testów walidacyjnych wykonanego oprogramowania i omówienie uzyskanych wyników.

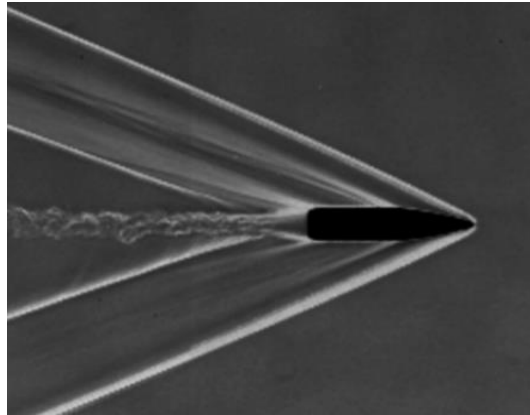
1. Zjawisko powstawania fali N i pomiar jej parametrów

1.1. Powstawanie i charakterystyka fali N

Tradycyjna broń palna wykorzystuje zamknięty ładunek wybuchowy, by wypchnąć kulę z lufy pistoletu. Podczas strzału mamy do czynienia z trzema rodzajami powstających fal akustycznych. Pierwszy generowany jest na skutek działania mechanizmów broni. Obejmuje on odgłosy działania elementów mechanizmu spustowego oraz ładowania amunicji. Dźwięki te posiadają niewielką amplitudę w porównaniu z innymi i wykrywane są w odległości kilku metrów od strzelającego.

Drugim rodzajem powstających dźwięków jest tzw. podmuch wylotowy, inaczej hałas balistyczny. Gorące, szybko rozprzestrzeniające się gazy powodują powstanie fali akustycznej wydostającej się z lufy, zwanej hałasem balistycznym. Fala rozchodzi się we wszystkich kierunkach, ale większość energii akustycznej emitowana jest w kierunku wskazanym przez lufę broni. Podmuch wylotowy powoduje akustyczną falę uderzeniową i krótki chaotyczny dźwięk [6]. Według doświadczeń naukowców trwa on około 3÷5 ms. Fala rozchodzi się z prędkością dźwięku (około 343 m/s przy 20°C). Fala akustyczna gazów nie jest jednak zbyt użytecznym źródłem do analizy. Dla większości broni palnych poziom dźwięku jest najsilniejszy w kierunku wskazywanym przez lufę i zmniejsza się wraz ze wzrostem kąta odchylenia od osi lufy. Fala może być również zasłonięta przez różne przeszkody między bronią a miejscem położenia czujnika pomiarowego. Ponadto niektóre rodzaje broni palnej można wyposażyć w tłumik, co zmniejsza znacznie poziom dźwięku [4].

Dlatego dużo bardziej użytecznym informacyjnie jest trzecie źródło fali akustycznej – fala dźwiękowa od pocisku poruszającego się z prędkością naddźwiękową. Obecnie zdecydowana większość broni wykorzystuje właśnie takie pociski. Naddźwiękowy pocisk porusza się zbyt szybko, by otaczające cząsteczki powietrza reagowały z normalną zależnością między ciśnieniem cząstek, a prędkością cząstek opisywaną przez zasady liniowej akustyki. Pocisk w ruchu naddźwiękowym generuje akustyczną falę uderzeniową (rys.1), która rozprzestrzenia się od toru lotu pocisku w formie stożka za nim, a jej front rozprzestrzenia się na zewnątrz z prędkością dźwięku [2].



Rysunek 1. Pocisk z falą uderzeniową [3].

Do opisu zjawiska powstawania akustycznej fali uderzeniowej od pocisku w ruchu naddźwiękowym wykorzystuje się liczbę Macha – jest to wielkość bezwymiarowa określająca stosunek prędkości obiektu poruszającego się w ośrodku do prędkości rozchodzenia się fali dźwiękowej w tym ośrodku.

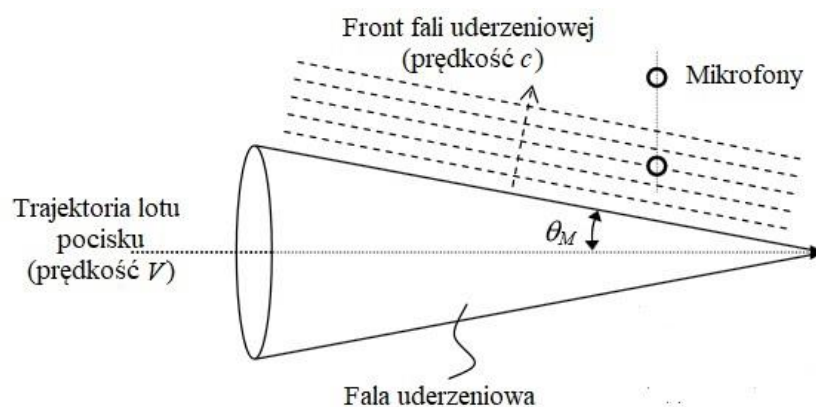
Liczba Macha wyrażona jest wzorem:

$$M = \frac{V}{c}$$

gdzie: M – liczba Macha, V – prędkość pocisku, c – prędkość dźwięku.

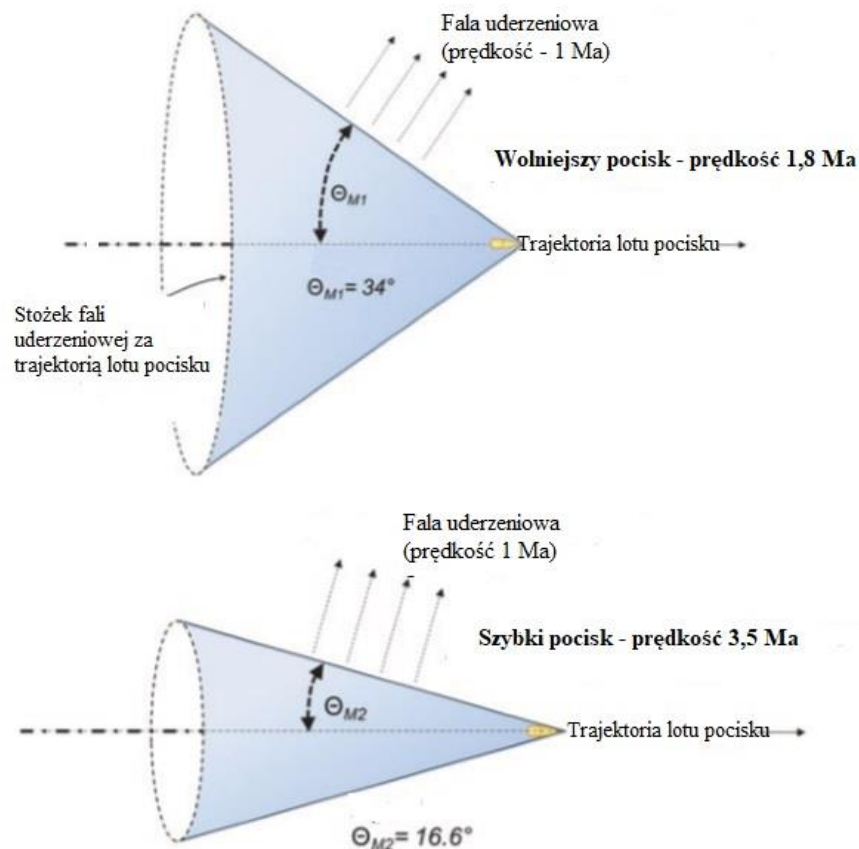
Wyróżnia się także kąt Macha θ_M :

$$\theta_M = \arcsin\left(\frac{1}{M}\right)$$



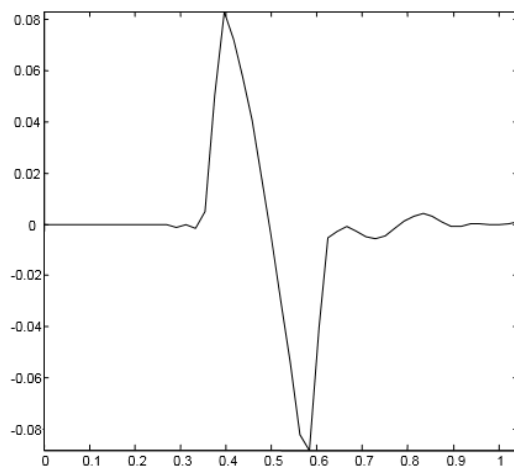
Rysunek 2. Schemat powstawania i rozchodzenia się fali uderzeniowej [4].

Dla bardzo szybkich pocisków liczba Macha jest duża i wtedy kąt rozwarcia stożka staje się mały, powodując propagację fali niemal prostopadle do trajektorii pocisku. Na przykład dla pocisku poruszającego się z prędkością 900 m/s, kąt rozwarcia stożka Macha wynosi około 20 stopni. Z drugiej strony, jeśli pocisk jest tylko nieznacznie szybszy od prędkości dźwięku, wtedy liczba Macha jest w przybliżeniu jednością, a kąt rozwarcia stożka Macha wynosi prawie 90 stopni, co powoduje, że fala rozchodzi się niemal równolegle do toru lotu pocisku (rys.3) [4].



Rysunek 3. Rozprzestrzenianie się i stożek fali uderzeniowej w zależności od prędkości pocisku [6].

To co charakteryzuje akustyczną falę uderzeniową od pocisku to bardzo szybki wzrost amplitudy ciśnienia do wartości maksimum, a następnie jej wolniejszy spadek do wartości minimum (podciśnienie). Zaburzenie ciśnienia tworzy kształt litery „N” z szybkim przejściem do maksimum związanym z przejściem czoła przedniej fali uderzeniowej, następnie niemal liniowym przejściem do minimum ciśnienia znacznie niższego od ciśnienia atmosferycznego, a następnie drugim skokiem ciśnienia spowodowanym przejściem tylnej fali uderzeniowej (rys.4) [1].



Rysunek 4. Przebieg fali N [4].

Według doświadczeń typowy kilkucentymetrowy pocisk generuje falę N o czasie trwania około 2 ms lub mniejszym [4].

1.2. Model matematyczny fali N

Przebieg fali uderzeniowej można opisać jako funkcję średnicy pocisku ϕ , prędkości V i odległości d między pozycją czujnika pomiarowego, a najbliższym punktem trajektorii pocisku. Odcinek ten jest prostopadły do toru pocisku. Przebieg fali N jest matematycznie określony za pomocą równań, które tworzą model Whithama [9,10]. Model Whithama można stosować w przypadku zaburzeń ośrodka przez ciała osiowosymetryczne, czyli np. pociski balistyczne [5]. Wyrażenie opisujące amplitudę fali N zgodnie z modelem Whithama jest następujące:

$$A = \frac{0,53P_0(M^2 - 1)^{1/8}\phi}{d^{3/4}l^{1/4}} \text{ [Pa] ,}$$

gdzie: P_0 – ciśnienie odniesienia ośrodka, l – długość pocisku, M – liczba Macha.

Wyrażenie opisujące czas trwania fali N zgodnie z modelem Whithama jest następujące:

$$T = \frac{1,82M\phi d^{1/4}}{l^{1/4}c(M^2 - 1)^{3/8}} \text{ [s] ,} \quad (1)$$

gdzie: c – prędkość dźwięku.

Ponadto wyróżnia się czas przybycia fali uderzeniowej do czujników pomiarowych τ_i :

$$\tau_i = \frac{s_i}{c},$$

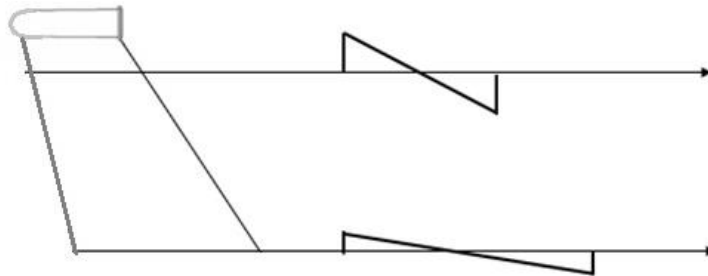
gdzie: s_i – odległość w linii prostej między źródłem zaburzenia, a czujnikiem pomiarowym w danym momencie.

Przebieg czasowy fali N można opisać następującą zależnością [5]:

$$y(t) = \begin{cases} A_i \left(1 - 2 \frac{t - \tau_i}{T}\right), & \tau_i \leq t \leq \tau_i + T \\ 0, & \text{dla innych przypadków} \end{cases}$$

Z przedstawionych wcześniej równań można wywnioskować, że wraz z wzrostem odległości między czujnikiem pomiarowym, a trajektorią pocisku, zmniejsza się amplituda fali N i zwiększa się jej czas trwania (rys.5).

Ponadto amplituda i czas trwania fali N wzrasta wraz ze zwiększeniem kalibru pocisku.



Rysunek 5. Szkic obrazujący zmianę kształtu fali N w zależności od odległości trajektorii pocisku od czujnika.

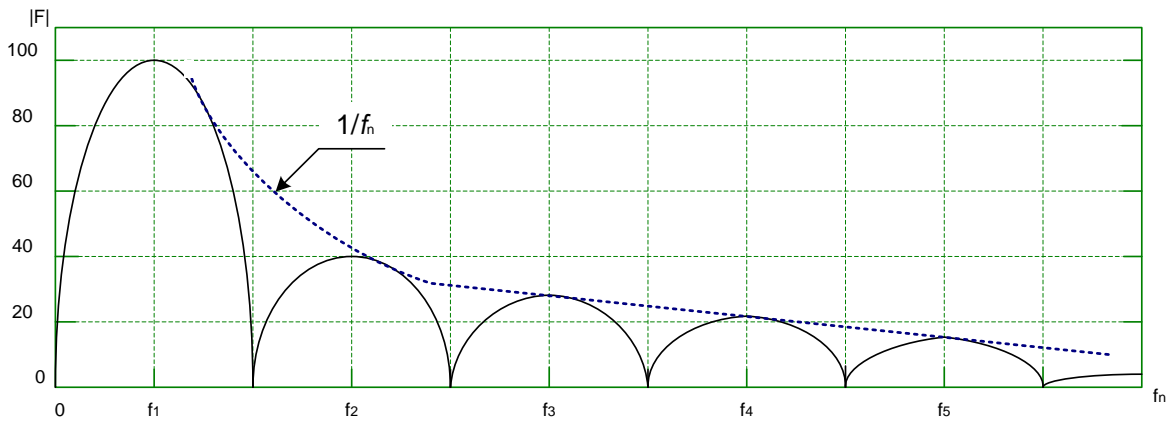
1.3. Charakterystyki częstotliwościowe fali N

Formuły modelu Whithama w polu swobodnym można wyrazić jako natężenie ciśnienia L i częstotliwości f [12]:

$$L(d) = 161,4 + 10 \lg \left(\frac{\phi^3}{l^{3/4}} \right) + 10 \lg \left(\frac{M^{9/4}}{(M^2 - 1)^{1/4}} \right) - 12,5 \lg \left(\frac{d}{d_0} \right)$$

$$f(d) = 176 \frac{(M^2 - 1)^{1/4}}{M^{3/4}} \cdot \frac{l^{1/4}}{\phi} \cdot \frac{1}{d^{1/4}}$$

Przykładowy przebieg charakterystyki częstotliwościowej fali N przedstawiono na rys.6:



Rysunek 6. Widmo amplitudowe fali N powstającej na skutek przelotu pocisku [11].

Na rys. 6 widać, że fala N posiada wolno zanikające widmo, co jest charakterystyczne dla sygnałów piłokształtnych. Widmo amplitudowe fali N opisane jest następującym równaniem:

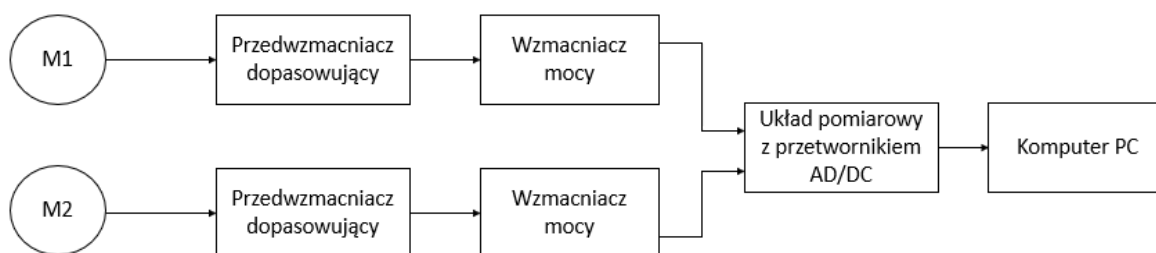
$$F(j\omega) = \frac{2A\psi}{\omega\psi} \left(\cos \omega\psi - \frac{\sin \omega\psi}{\omega\psi} \right),$$

gdzie: poprzez $2\psi = T$ oznaczono czas trwania fali N, a ω jest częstotliwością kątową [13].

Wykreślona na charakterystyce amplitudowej widma fali N obwiednia pokazuje, że maksymalne amplitudy kolejnych harmonicznnych fali N są odwrotnie proporcjonalne do częstotliwości f_n . Świadczy to, że szybkość spadku amplitud nie zależy od czasu trwania T fali N, lecz od czasu narastania zbocza rozpatrywanego zaburzenia [13].

1.4. Pomiar fali N

Pomiar fali N polega na umieszczeniu czujników pomiarowych w zasięgu fali uderzeniowej. Czujnik akustyczny (np. mikrofon) rejestruje zmiany ciśnienia i przetwarza na sygnał elektryczny. Ten następnie jest wzmacniany i podawany na przetwornik A/C układu pomiarowego. Do analizy i prezentacji wyników pomiaru wykorzystuje się komputer PC [1].

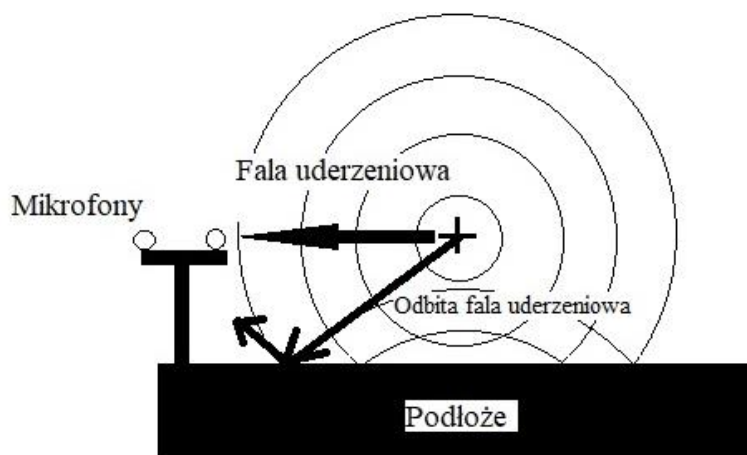


Rysunek 7. Schemat układu pomiarowego fali uderzeniowej [1].

W zagadnieniach lokacji akustycznej obiektów ważną kwestią jest dobór odpowiedniej ilości czujników pomiarowych. Typowy problem lokacji akustycznej obejmuje oszacowanie równania trajektorii pocisku, a następnie azymutu, wysokości i odległości względem zdefiniowanego punktu odniesienia. Para mikrofonów o znanym położeniu może określić czas nadejścia fali uderzeniowej pocisku. Za pomocą trzech mikrofonów można oszacować wysokość i odległość trajektorii pocisku w stosunku do znanej lokalizacji mikrofonów, ale orientacja pocisku i prędkość nadal nie są jednoznaczne. Jeżeli znana jest prędkość początkową pocisku, wtedy można oszacować liczbę Macha, kąt rozwarcia stożka i dalej trajektorię lotu. Zasadniczo niezbędny zestaw do lokacji pocisku powinien składać się z minimum czterech przestrzennie rozlokowanych mikrofonów [4].

Oszacowanie typu użytej broni i amunicji na podstawie fali akustycznej jest jednak dużo trudniejsze. Dwa zapisy strzałów z tej samej broni mogą się znacznie różnić, chyba że mikrofon znajduje się w ustalonym miejscu w stosunku do broni palnej, a otoczenie akustyczne jest takie samo [2].

Warunki pomiaru nigdy nie są jednak idealne (jednorodne). Jeśli w pobliżu znajdują się stałe powierzchnie, przechodzący stożek fali uderzeniowej zostanie częściowo pochłonięty i częściowo odbity przez powierzchnie. W związku z tym mikrofon odbierze zarówno oryginalną falę uderzeniową, jak i odbitą falę uderzeniową z opóźnieniem (rys.8).



Rysunek 8. Powstawanie odbitej fali uderzeniowej.

Najlepszym otoczeniem dla prowadzenia badań jest wykonanie pomiarów w komorze bezechowej, która pochłania wszystkie odbite dźwięki i nie dopuszcza zewnętrznych hałasów (rys.9).

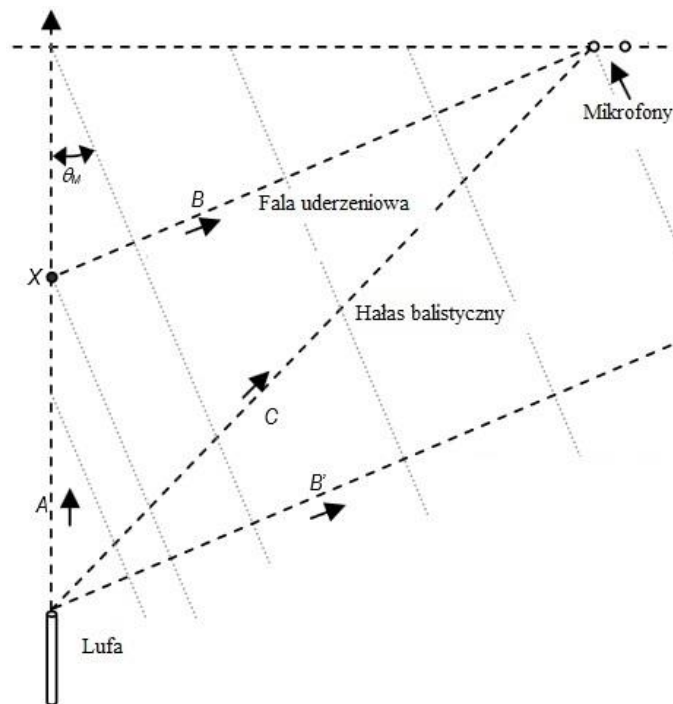


Rysunek 9. Komora bezechowa.

<http://www.lat.agh.edu.pl/pomieszczenia/komora-bezechowa/>

Kolejnym czynnikiem zakłócającym idealny pomiar jest opór powietrza. Jeżeli pocisk przebywa drogę większą niż kilkadziesiąt metrów, trzeba wziąć pod uwagę skutki tarcia powietrza. Pocisk zostaje spowolniony i następuje przekształcenie jego energii kinetycznej w akustyczną falę uderzeniową. Skutkuje to tym, że granica fali uderzeniowej jest w istocie wypukła, nie prosta [1].

Geometrię pomiaru fali uderzeniowej przedstawia rysunek 10.

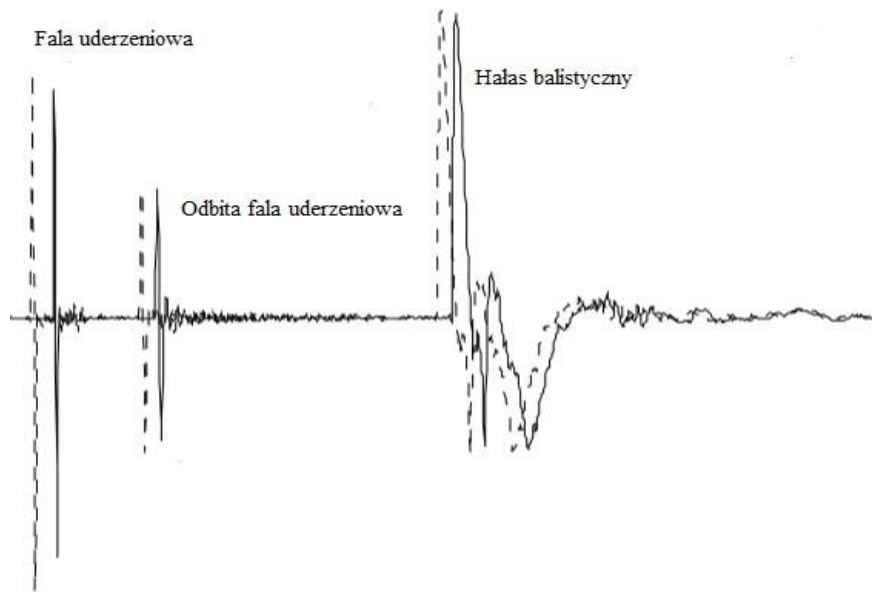


Rysunek 10. Geometria pomiaru fali uderzeniowej [4].

Pocisk po wystrzeleniu z lufy porusza się po drodze A. Fronty czoła fal uderzeniowych propagują w kierunku mikrofonów gdy pocisk osiągnie punkt X – w tym miejscu prosta prostopadła do frontu czoła fali przecina trajektorię lotu pocisku. Punkt X nazywany jest punktem źródłowym zaburzenia. Natomiast miejsce detekcji fali N przez mikrofon nazywany jest punktem detekcji zaburzenia. Całkowity czas od wystrzelenia pocisku do detekcji, zawiera czas przelotu pocisku z lufy do punktu X oraz czas propagacji fali uderzeniowej z prędkością dźwięku do mikrofonów wzdłuż drogi B. Geometrycznie ten czas jest równy czasowi propagacji wzdłuż drogi B' [4].

Odbita fala uderzeniowa rozchodzi się w tym samym azymucie co fala uderzeniowa, ale po dłuższej drodze – do podłoża, a następnie dopiero do mikrofonów. Pojawi się zatem w punkcie detekcji później. Hałas balistyczny propaguje po drodze C – bezpośrednio z lufy do mikrofonów. Droga ta jest dłuższa od drogi propagacji fali uderzeniowej [4].

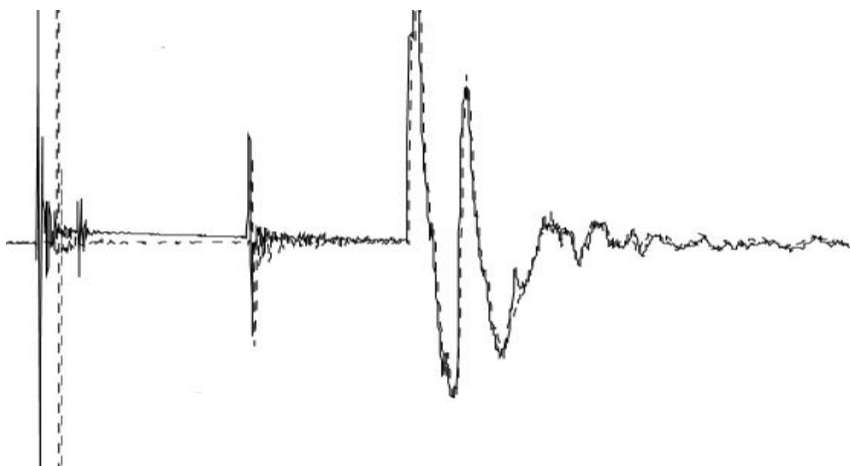
Wyniki przykładowych rejestracji fali N są przedstawione na rysunku 11.



Rysunek 11. Wyniki pomiaru fali N [2].

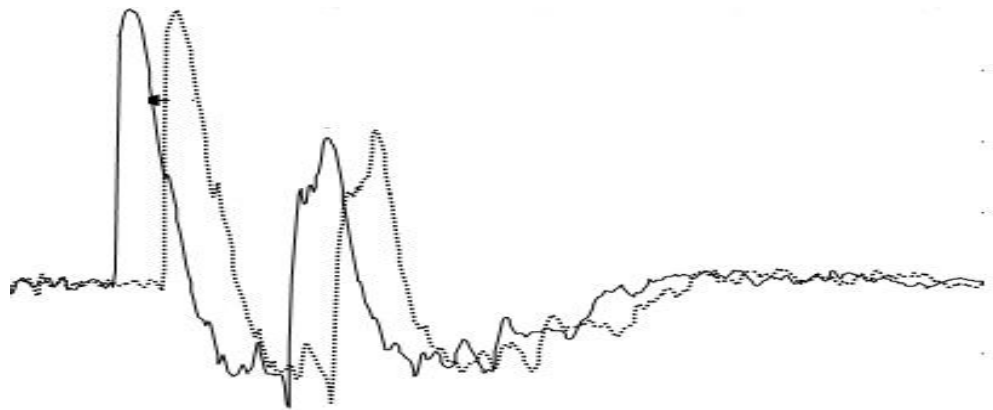
Z analizy rys. 11 wynika, że odbita fala ma bardzo podobny przebieg jak główna fala uderzeniowa, jednakże ma mniejszą amplitudę spowodowaną częściowym pochłonięciem energii fali przez podłoże oraz dłuższą drogę propagacji. Hałas balistyczny może osiągać większe amplitudy niż fala N i ma znacznie mniej regularny przebieg.

Gdy odległość mikrofonów względem trajektorii pocisku będzie mniejsza, to rezultaty pomiaru zjawiska prezentują się jak na rysunku 12.



Rysunek 12. Pomiary fali N przy bliższej odległości mikrofonu od pocisku [2].

Przybliżenie mikrofonów w kierunku trajektorii pocisku powoduje w rezultacie zarejestrowanie bardzo silnej fali uderzeniowej o wyższej amplitudzie ciśnienia, w odniesieniu do nieznacznego wzrostu amplitudy hałasu balistycznego i fali odbitej. Znaczne oddalenie mikrofonów od trajektorii pocisku (rys.13) spowoduje, że fala N będzie stopniowo tracić swój charakterystyczny kształt i stanie się zaburzeniem ciśnienia o kształcie „S”. Także szумы otoczenia (zakłócenia) będą coraz większe w porównaniu do zaburzenia, aż fala stanie się niezauważalna [4].



Rysunek 13. Rejestracja fali w znacznym oddaleniu mikrofonów [4].

1.5. Wpływ warunków środowiska na pomiary parametrów fali N

Fala uderzeniowa, podobnie jak inne fale fizyczne, podlega odbiciu, osłabieniu, absorpcji, dyfrakcji i innym modyfikacjom podczas jej propagacji. Mikrofon lub inny czujnik akustyczny zarejestruje fale docierające ze źródła, jak i fale docierające później z innych kierunków z powodu odbić i rozproszenia. Jeżeli skutki odbić i propagacji w atmosferze nie są jednoznacznie uwzględnione przez zaimplementowane modele i algorytmy w systemie lokacji obiektów, poprawność działania takiego systemu powinna być starannie sprawdzona w warunkach rzeczywistych.

Propagacja fali dźwiękowej może znacznie różnić się ze względu na warunki atmosferyczne, dyfrakcję wokół przeszkadzających obiektów, odbicia od ziemi i innych powierzchni. Co więcej, na propagację dźwięku może wpływać wiatr, zmiany temperatury oraz pochłanianie atmosferyczne zależne od częstotliwości. W związku z tym, że te specyficzne warunki środowiskowe będą znacznie się różnić

w zależności od lokalizacji, nie jest możliwe zbudowanie jednego, kompleksowego modelu matematycznego, mającego ogólne zastosowanie. Modele empiryczne są zatem wymagane w celu uwzględnienia warunków środowiskowych. Zwykle, słyszalne dźwięki mieszczą się w zakresie ciśnienia, który jest dobrze modelowany liniowymi równaniami różniczkowymi fal, ale fale uderzeniowe spowodowane naddźwiękowymi pociskami zachowują się w powietrzu nieliniowo. Możliwe jest przewidywanie właściwości akustycznych danego środowiska przy użyciu standardowych metod matematycznych, jednak właściwości absorpcji, tłumienia i odbicia od obiektów muszą być znane dla liniowych i nieliniowych propagacji. Typowe dźwięki obejmują zakres częstotliwości od poniżej 10 Hz do ponad 40 kHz. Długości fali napotymane w powietrzu mogą zatem wahać się od ponad 30 m przy niskich częstotliwościach, do mniej niż 0,01 m dla najwyższych częstotliwości. Zakres długości fal jest duży, oznacza to że właściwości dyfrakcyjne i absorpcyjne będą znacznie się różnić w zależności od spektrum źródła dźwięku [2].

Opady atmosferyczne zwykle nie mają znaczącego wpływu na rejestrację fal dźwiękowych. Z kolei wilgotność, wiatr i zmiany temperatury mają wpływ na wartość rejestrowanych parametrów fal dźwiękowych, a także generują hałas akustyczny.

Duży wpływ na pomiar parametrów fal dźwiękowych ma wiatr. Fala dźwiękowa jest przenoszona w ruchomej masie powietrza. Poprzez działanie wiatru ruch czoła fali uderzeniowej będzie składał się z sumy wektorowej sferycznie rozwijającego się wektora dźwięku oraz wektora wiatru. Efekt wiatru można postrzegać jako przesunięcie punktu początkowego propagacji dźwięku. Czoło fali jest systematycznie przesuwane przez wiatr, tak że po dotarciu fali do mikrofonów, przesunięte zostaje pozorne położenie źródła dźwięku i trajektoria pocisku. Prędkość wiatru jest zwykle wyższa na wysokości i niższa na dole. To powoduje, że fale dźwiękowe propagujące z wiatrem są wygięte w górę, a te które propagują się pod wiatr, są wygięte w dół. Chociaż prędkości wiatru są dużo niższe w porównaniu z prędkością dźwięku, mają one wpływ, gdyż zmiana wiatru powoduje zależną od kierunku zmianę prędkości dźwięku, a co za tym przesunięcie częstotliwości, podobne do efektu Dopplera [2].

Wpływ nierównomiernej temperatury może również być znaczący. Temperatura powietrza w atmosferze nie jest jednorodna. W dzień, szczególnie w miesiącach letnich powierzchnia ziemi jest cieplejsza niż powietrze. W tej sytuacji propagacja

dźwięku ma tendencje do pochylania się nieco w górę ze względu na temperaturę. Czoło fali w ciepłym powietrzu w pobliżu powierzchni ziemi porusza się szybciej niż czoło fali w chłodniejszym powietrzu wyżej nad ziemią. Zimą lub w nocy, gdy temperatura powierzchni ziemi jest niższa, fale dźwiękowe są pochylane w dół. Połączone efekty wiatru i zmian temperatury mogą spowodować, że poziomy dźwięku mierzone w pewnej odległości od źródła będą bardzo różne od przewidywań opartych na rozkładzie geometrycznym i rozważaniach dotyczących absorpcji atmosferycznej. Różnice te mogą wynosić 20 dB lub więcej na odległościach kilkuset metrów [2].

Kolejnym czynnikiem jest powierzchnia ziemi i przeszkody. Odgłosy wystrzałów rozprzestrzeniających się nad ziemią ulegają osłabieniu poprzez straty energii akustycznej spowodowanej rozpraszaniem. Gładkie i twarde podłoże generalnie będzie mniej absorbowało niż szorstkie jak np. roślinność. Wyższe częstotliwości (krótsze fale) są tłumione bardziej niż niższe częstotliwości. Pomiar na obszarach zalesionych pokazuje, że absorpcja i rozpraszanie mogą osiągnąć znaczne tłumienie. Problemem może również być tłumienie poprzez przeszkodę zasłaniającą linię widzenia między źródłem, a czujnikiem akustycznym [2].

Wilgotność względna powietrza powoduje pochłanianie dźwięku zależne od częstotliwości z powodu relaksacji termicznej cząsteczek. Stwierdzono, że tłumienie wzrasta monotonicznie wraz ze wzrostem częstotliwości i jest największe dla wilgotności względnej w zakresie 10-30% [2].

Znaczenie czynników atmosferycznych zależy od rozważanej sytuacji. Na przykład dla źródła dźwięku w pobliżu ziemi i odległości mniejszej niż 25 m od czujników akustycznych w miarę czystym terenie od przeszkód, wskazane wyżej czynniki nie mają istotnego wpływu. Jednak dla mikrofonów rozlokowanych w odległości powyżej 100 m od wystrzałów, należy już uwzględniać warunki powierzchni, przeszkody i czynniki atmosferyczne.

2. Projekt i implementacja algorytmu do wyznaczania parametrów czasowo częstotliwościowych fali N

2.1. Ogólne informacje o programie

W związku z planowanymi oraz już przeprowadzonymi przez pracowników z Zespołu Radioelektroniki Wojskowej Akademii Technicznej doświadczeniami, które miały na celu badanie zaburzeń ośrodka generowanych przez pociski podczas strzelań, zaistniała potrzeba wykonania programu komputerowego, za pomocą którego na podstawie zarejestrowanych danych można byłoby wyznaczyć parametry fali N. Informacje te mogą później posłużyć do tworzenia modeli określających np. położenie strzelca lub rodzaj użytej broni i amunicji. Głównymi wymaganiami dotyczącymi programu były: niezawodność, sprawność i uniwersalność. W celu poprawy komfortu pracy, program wyposażono w interfejs graficzny oraz skompilowano do postaci wykonywalnej. Kod napisano w języku *Python 3*, używając do tego środowiska *Python Spyder*. Język *Python* wybrano z powodu jego funkcjonalności w analizie danych, światowe trendy w stosowaniu przez firmy oraz przejrzystej składni. Zastosowanym środowiskiem graficznym jest *Tkinter*. W napisaniu programu bardzo pomocne okazały się pozycje literatury [7] i [8].

2.2. Główne funkcjonalności programu

Opracowany program charakteryzują następujące funkcjonalności:

- wczytywanie danych z pliku z rozszerzeniem *.txt* lub *.csv*,
- wybór danych z kanału pomiarowego jednym kliknięciem,
- znalezienie wartości amplitudy ciśnienia fali N,
- oszacowanie czasu trwania fali N,
- oszacowanie wartości częstotliwości pierwszej harmonicznej fali N,

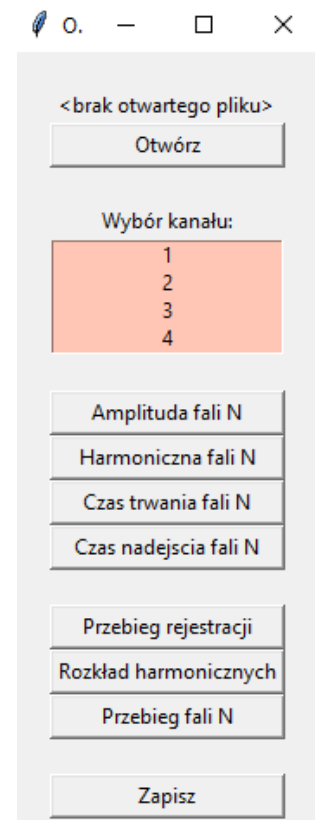
- oszacowanie czasu przyścia fali N do mikrofonów (dla pomiarów wielokanałowych),
- rysowanie przebiegów czasowych rejestracji (przydatne w celu szybkiego podglądu czy dane zawierają cenne informacje),
- rysowanie przebiegów czasowych fali N,
- rysowanie histogramu charakterystycznych częstotliwości harmoniczných
- zapis szacowanych parametrów, a także nazwy pliku z którego zostały pobrane dane, do wybranego pliku z rozszerzeniem .csv lub .txt.

2.3. Opis działania programu

Po wczytaniu danych z wybranego pliku pomiarowego użytkownik wskazuje dane z kanału pomiarowego, a następnie korzystając z menu głównego programu i przechodzi do analizy wczytanych danych, korzystając z funkcjonalności programu. Program bazuje na definicjach funkcji, które mogą być wielokrotnie wykorzystane. Niektóre z nich są podpięte do przycisków interfejsu.



Rysunek 14. Ogólny schemat działania programu.



Rysunek 15. Menu główne programu.

Naciśnięcie przycisku *Otwórz* uruchamia funkcję *openFile()*, która wykonuje następujące działania:

- wyświetla okno wyboru pliku, przypisuje nazwę pliku do zmiennej;
- otwiera plik, wczytuje wierszami jego zawartość (funkcja *readlines()*) i zamyka go;
- w przypadku niepowodzenia operacji wyświetla komunikat o błędzie;
- wczytuje wartość częstotliwości próbkowania z pierwszego wiersza;
- przy pomocy pętli *for* zamienia spójniki w danych, tak by zarejestrowane dane mogły być interpretowalne przez program, wskazuje spójnik oddzielający dane z poszczególnych kanałów i wczytuje dane kolumnami;
- przypisuje zmiennej *fileloaded* wartość *True*; wartość ta jest wykorzystywana w celu informowania, czy plik został poprawnie wczytany.

Po poprawnym wczytaniu pliku w menu głównym pojawia się jego pełna nazwa (dzięki poleceniu *selectionLabel.config()*).

Kod opisanej części programu:

```
def openFile():
    global filename
    global text
    global SampleFreq
    global NumChan
    global NumSampPerBlock
    global NumSampPerChan
    global column
    global fileLoaded
    global selectionLabel

    filename = tkinter.filedialog.askopenfilename(initialdir = "/", title = "Otwórz plik:"
        , filetypes = (("Pliki CSV", "*.csv"), ("Pliki tekstowe", "*.txt")))
    if not filename:
        tkinter.messagebox.showwarning("Niepowodzenie", "Nie wybrano pliku!")
        return

    plik = open(filename)
    text = plik.readlines()
    plik.close()

    SampleFreq = float(text[0][17:-1].replace(',', '.'))

    text = text[13:]
    column = [[] for i in range(4)]

    for x in text:
        x = x.replace(',', '.')
        fields = x.split('; ')
        for i in range(4):
            column[i].append(float(fields[i]))

    fileLoaded = True
    selectionLabel.config(text=filename)
```


W przypadku, gdy nie zostanie wybrany żaden plik do analizy, a mimo to używać się będzie przycisków funkcjonalnych programu, program będzie wyświetlał komunikat o błędzie i konieczności wyboru pliku. Odpowiada za to zdefiniowana zmienna *fileloaded*, której domyślnie przypisano wartość *False*. Zmienia ona swoją wartość na *True* dopiero po poprawnym załadowaniu pliku.

Kod:

```
def fileNotLoadedException():
    global fileloaded
    if not fileloaded:
        tkinter.messagebox.showwarning("Nieprawidłowe polecenie", "Najpierw wybierz plik!")
        return True
    return False
```

Przy każdej funkcji, która jest bezpośrednio zdefiniowana do przycisku, sprawdzana jest na początku wartość zmiennej *fileloaded*.

Kod na początku każdej funkcji:

```
if fileNotLoadedException():
    return
```

Aby móc dalej efektywnie korzystać z programu, w menu głównym należy zaznaczyć kanał z którego dane mają być analizowane. W przeciwnym przypadku, przy każdorazowym przyciśnięciu dowolnego przycisku (z wyjątkiem przycisku *Otwórz*), pojawi się komunikat o niepowodzeniu operacji i konieczności zaznaczenia kanału z danymi do analizowanego.

Funkcja odpowiadająca za komunikat o błędzie:

```
def noSelectionException():
    global selection
    if not selection:
        tkinter.messagebox.showwarning("Nieprawidłowe polecenie", "Najpierw wybierz kanał!")
        return True
    return False
```

Kod realizujący poprawne działanie programu po wyborze kanału jest następujący:

```
def onselect(event):
    global selection
    widget = event.widget
    selection = widget.get(int(widget.curselection()[0]))
```

Następnie program przypisuje do zmiennej *selection* odpowiednią wartość w zależności od numeru wskazanego kanału. Zmienna *selection* informuje program, który kanał z danymi został wskazany do analizy.

Po wczytaniu danych z wybranego kanału, w celu dalszej analizy należy wykonać następne polecenia w dowolnej kolejności i ilości, gdyż kolejne przyciski z funkcjonalnościami zdefiniowano jako funkcje niezależne.

Po naciśnięciu przycisku *Amplituda fali N*, uruchamiana jest funkcja *showMax()*, która wyświetla wartość amplitudy fali N, korzystając z polecenia *max()*.

Kod do wywołania działania funkcji *Amplituda fali N* jest następujący:

```
def showMax():
    global selection
    global column
    if fileNotLoadedException():
        return
    if noSelectionException():
        return
    tkinter.messagebox.showinfo("Amplituda", "Wartość amplitudy fali N wynosi:
    " + str(max(column[int(selection)-1])))
```

Po naciśnięciu przycisku *Czas trwania fali N*, uruchamiana jest funkcja *showPeriod()*, która wykonuje następujące działania:

- znajduje indeks pierwszej i ostatniej próbki tworzącej falę N wykorzystując funkcje *findBegin* i *findEnd*;
- oblicza czas trwania fali N w następujący sposób: znajduje czas między dwoma kolejnymi próbkami poprzez odwrócenie wartości częstotliwości próbkowania i mnoży przez ilość próbek tworzących falę N (różnica między numerem indeksu ostatniej próbki i pierwszej);
- dla czytelności mnoży czas trwania przez 1000 i zaokrągla do trzech miejsc po przecinku poleceniem *round()*;
- wyświetla wynik.

Efektem końcowym jest wyświetlenie wartości czasu trwania fali N.

Kod do wywołania działania funkcji *Czas trwania fali N* jest następujący:

```
def showPeriod():
    global SampleFreq
    global selection
    global column
    if fileNotLoadedException():
        return
    if noSelectionException():
        return
    index = int(selection)-1
    period = 1 / SampleFreq * (findEnd(column[index],
                                         column[index].index(min(column[index])))
    - findBegin(column[index], column[index].index(max(column[index])))
    period = period*1000
    period = round(period, 3)
    tkinter.messagebox.showinfo("Czas trwania", "Czas trwania fali N wynosi: "
                                + str(period) + " ms")
```

W tej funkcji, ale też niektórych następnych, zagnieżdżone są funkcje *findBegin* oraz *findEnd*. Kod funkcji *findBegin*:

```
def findBegin(channel, index):
    while index >= 0 and channel[index] > channel[index-1]:
        index -= 1
    return index
```

Funkcja *findBegin* zwraca indeks (docelowo indeks próbki początkowej fali N). Wykorzystuje ona pętlę *while*. Zaczynając od określonej wartości (docelowo indeksu *max*), idąc indeksami w dół, szuka indeksu próbki która jest mniejsza od próbki o indeksie o jeden mniejszym. Oznacza to, że we wskazanym miejscu znajduje się ekstremum lokalne, które jest początkiem fali N. Opracowany fragment kodu uwzględnia następującą własność fali N, iż pierwszy wzrost ciśnienia jest szybki, gwałtowny i kończy się dopiero na wartości maksymalnej.

Kod funkcji *findEnd*:

```
def findEnd(channel, index):
    while index < len(channel) and channel[index] <= 0:
        index += 1
    return index
```

Funkcja *findEnd* jest podobna w działaniu do funkcji *findBegin*. Również wykorzystuje pętlę *while*. Szuka docelowo indeksu próbki końcowej fali N w następujący sposób: zaczynając od określonej wartości (docelowo indeksu o wartości minimalnej, posuwa

się z indeksami do przodu, szukając próbki która przekroczy wartość zero. W tym miejscu kończy się czas trwania fali N.

Po naciśnięciu przycisku *Rozkład harmoniczných* uruchamiana jest funkcja *showFFT*, która wykonuje następujące działania:

- znajduje za pomocą funkcji *findBegin* i *findEnd* indeksy pierwszej i ostatniej próbki tworzącej falę N;
- za pomocą polecenia *NumPy fft()* liczy szybką transformatę Fouriera dla fali N;
- wykorzystuje polecenie *len()* dla znalezienia liczby próbek tworzących falę N;
- tworzy poleceniem *linspace()* wektor liczb (częstotliwości) od zera do końcowej wartości częstotliwości próbkowania *SampleFreq* o ilości liczb równej obliczonej funkcją *len()*. Wektor ten tworzy opis osi x;
- tworzy wykres charakterystycznych częstotliwości harmoniczných poleceniem *plt.bar* (wykres słupkowy z biblioteki *Matplotlib*), na osi y umieszczając bezwzględne wartości obliczonej szybkiej transformaty Fouriera (polecenie *abs*), odpowiednio przycinając wektor tych liczb do wartości mających istotne znaczenie. Poleceniem *width* poprawia czytelność wykresu;
- opisuje tekstem osie;
- dodaje siatkę do wykresu.

Efektem końcowym działania przycisku *Rozkład harmoniczných* jest wyświetlenie histogramu charakterystycznych częstotliwości harmoniczných fali N wyznaczonych za pomocą szybkiej transformaty Fouriera.

Kod:

```
def showFFT():
    global SampleFreq
    global column
    global selection
    if fileNotLoadedException():
        return
    if noSelectionException():
        return

    index = int(selection) - 1
    beg, end = findBegin(column[index], column[index].index(max(column[index]))),
    + findEnd(column[index], column[index].index(min(column[index])))
    fu = np.fft.fft(column[index][beg:end])
    n=len(fu)
    f=np.linspace(0, SampleFreq, n)
    plt.bar(f[:n // 6], np.abs(fu)[:n // 6] * 1 / n, width=1500)
    plt.ylabel("Amplitude")
    plt.xlabel("Frequency [Hz]")
    plt.grid()
    plt.show()
```

Naciśnięcie przycisku *Przebieg rejestracji* powoduje uruchomienie funkcji *showChartwst()*, która rysuje wykres wszystkich danych wybranego przez nas kanału, w tym próbek które nie wchodzą w skład fali N, zawartych w interesującym nas pliku. Kod:

```
def showChartwst():
    global selection
    global column
    global SampleFreq
    if fileNotLoadedException():
        return
    if noSelectionException():
        return
    index = int(selection)-1

    y= column[index]

    plt.ylabel("Amplitude")
    plt.xlabel("Liczba próbek")

    plt.plot(y)
    plt.show()
```

Naciśnięcie przycisku *Przebieg fali N* uruchamia funkcję *showChart()*, której przebieg jest następujący:

- znajduje indeks próbki początkowej i końcowej fali N;
- przypisuje do wektora *y* wartości próbki wchodzące w skład fali N, dodatkowo także dodaje po kilkadziesiąt próbek na początku i na końcu przebiegu fali N;
- oblicza czas rejestracji pierwszej i ostatniej próbki wchodzącej w skład wykresu i na ich podstawie tworzy poleceniem *linspace* wektor o ilości elementów równej liczbie próbek wchodzących w skład wykresu, obliczonej za pomocą polecenia *len()*;
- dodaje opisy osi;
- rysuje wykres.

Kod:

```
def showChart():
    global selection
    global column
    global SampleFreq
    if fileNotLoadedException():
        return
    if noSelectionException():
        return
    index = int(selection)-1
    beg, end = findBegin(column[index], column[index].index(max(column[index])))
    , findEnd(column[index], column[index].index(min(column[index])))

    y= column[index][beg-50:end+100]
    z=len(y)

    x=np.linspace(1000*(1/SampleFreq)*(beg-50), 1000*(1/SampleFreq)*(end+100)
    , z)
    plt.ylabel("Amplitude")
    plt.xlabel("Time [ms]")

    plt.plot(x,y)
    plt.show()
```

Efekt działania przycisku *Przebieg fali N* jest rysowanie jej przebiegu.

Przycisk *Czas nadejścia fali N* uruchamia funkcję *showTime()*, która wykonuje następujące kroki:

- znajduje indeks pierwszej próbki fali N funkcją *findBegin*;
- oblicza czas jej rejestracji, odwracając wartość częstotliwości próbkowania i mnożąc ją przez indeks próbki;
- zaokrągla wartość do trzech miejsc po przecinku i wyświetla w formie milisekund.

Kod:

```
def showTime():
    global selection
    global column
    global SampleFreq
    if fileNotLoadedException():
        return
    if noSelectionException():
        return

    index = int(selection) - 1
    beg = findBegin(column[index], column[index].index(max(column[index])))
    time=1/SampleFreq*beg
    time=time*1000
    time=round(time,3)
    tkinter.messagebox.showinfo("Czas nadejścia",
    "Fale N zarejestrowano po upływie: " + str(time) + " ms")
```

Efektom działania przycisku *Czas nadejścia fali N* jest podanie czasu zarejestrowania jej początku.

Naciśnięcie przycisku *Harmoniczna fali N* spowoduje rozpoczęcie działania funkcji `showFreq()`, która wykonuje kolejne działania:

- znajduje indeks pierwszej i ostatniej próbki fali N;
- liczy szybką transformatę Fouriera dla wektora próbek fali N i przypisuje ją do wektora `fu1`;
- liczy liczbę elementów wektora `fu1`, dzieli ją na pół i zaokrągla funkcją `round()`;
- liczy część rzeczywistą elementów wektora powstałego po transformacie zredukowanego do policzonej wcześniej liczby elementów;
- za pomocą pętli `while` szuka wartości maksymalnej transformaty i indeksu tej wartości w wektorze `fu1`;
- liczy częstotliwość danej próbki mnożąc znaleziony indeks próbki przez odwróconą wartość częstotliwości próbkowania podzieloną przez ilość elementów wektora `fu1` całej transformaty;
- zaokrągla wynik do dwóch miejsc po przecinku i wyświetla wartość częstotliwości.

Kod:

```
def showFreq():
    global SampleFreq
    global column
    global selection
    global czestotliwosc
    if fileNotLoadedException():
        return
    if noSelectionException():
        return

    index = int(selection) - 1
    beg, end = findBegin(column[index], column[index].index(max(column[index])))
    findEnd(column[index], column[index].index(min(column[index])))
    fu1 = np.fft.fft(column[index][beg:end])
    n1=round((len(fu1))/2)
    fu2=abs(fu1[:n1])
    m=1
    maxii=0
    while m<n1:
        if fu2[m]>maxii:
            maxii=fu2[m]
            maxi2=m
        m=m+1
    czestotliwosc=maxi2*(SampleFreq/(2*n1))
    czestotliwosc=round(czustotliwosc,2)

    tkinter.messagebox.showinfo("F0", "Wartość częstotliwości pierwszej
                                   harmonicznej wynosi " + str(czustotliwosc) + " Hz")
```


Efekt działania przycisku *Harmoniczna fali N* jest obliczenie i wyświetlenie wartości częstotliwości pierwszej harmonicznej (częstotliwości dominującej) fali N.

Naciśnięcie przycisku *Zapisz* spowoduje uruchomienie funkcji *saveToFile()*.

Funkcja ta:

- liczy parametry fali N: wartość amplitudy, częstotliwości dominującej, czas trwania i czas nadejścia fali w taki sam sposób jak funkcje *showMax()*, *showFreq()*, *showPeriod()* i *showTime()*. Jest to powtórzenie kodu, jednakże dzięki temu program jest bardziej użyteczny. Można wgrać plik do analizy, wybrać kanał i od razu przyciskiem *Zapisz* zapisać w wybranej lokalizacji interesujące parametry, bez sprawdzania ich wcześniej. Jest to przydatne w obsłudze wielu plików;
- wyświetla okno dialogowe wyboru pliku do którego zapisać dane, w przypadku niewybrania pliku pojawia się komunikat o niepowodzeniu;
- otwiera plik do zapisu poleceniem *open*. Argument 'a' informuje o tym, że nie nadpisujemy danych w pliku, tylko dopisujemy na jego końcu;
- zapisuje dane funkcją *readlines()* i zamyka plik.

Efekt działania przycisku *Zapisz* jest zapis danych do wybranego wcześniej pliku.

Kod:

```
def saveToFile():
    global column
    global selection
    global SampleFreq
    global czestotliwosc
    global filenameopen
    if fileNotLoadedException():
        return
    if noSelectionException():
        return

    index = int(selection) - 1

    maxi = str(max(column[index]))

    period = 1 / SampleFreq * (findEnd(column[index], column[index].index(min(column[index])))
    - findBegin(column[index], column[index].index(max(column[index]))))
    period=period*1000
    period=round(period,3)

    beg, end = findBegin(column[index], column[index].index(max(column[index]))),
    findEnd(column[index], column[index].index(min(column[index])))
    fu1 = np.fft.fft(column[index][beg:end])
    n1=round((len(fu1))/2)
    fu2=abs(fu1[:n1])
    m=1
    maxii=0
    while m<n1:
        if fu2[m]>maxii:
            maxii=fu2[m]
            maxi2=m
        m=m+1
    czestotliwosc=maxi2*(SampleFreq/(2*n1))
    czestotliwosc=round(czestotliwosc,2)
```


Kolejna część kodu:

```
beg = findBegin(column[index], column[index].index(max(column[index])))
time=1/SampleFreq*beg
time=time*1000
time=round(time,3)

filename = tkinter.filedialog.asksaveasfilename(initialdir="/", title="Zapisz plik:",
                                                filetypes=(("Pliki CSV", "*.csv"), ("Pliki tekstowe", "*.txt")))
if not filename:
    tkinter.messagebox.showwarning("Niepowodzenie", "Nie wybrano pliku!")
    return

plik = open(filename, 'a')
plik.writelines('\n' + str(filenameopen) + ', ' + 'kanał ' + str(selection) + ', ' + str(maxi) + ', ' + str(period) + ' ms,')
plik.close()
```

Opisane powyżej funkcje są przypisane do interfejsu menu głównego programu w następujący sposób:

```
maxButton = tkinter.Button(fFrame, text="Amplituda fali N", command=showMax, width=18, height=1)
fButton = tkinter.Button(fFrame, text="Harmoniczna fali N", command=showFreq, width=18, height=1)
periodButton = tkinter.Button(fFrame, text="Czas trwania fali N", command=showPeriod, width=18, height=1)

chartButton = tkinter.Button(chartFrame, text="Przebieg fali N", command=showChart, width=18, height=1)
fftButton = tkinter.Button(chartFrame, text="Rozkład harmoniczných", command=showFFT, width=18, height=1)

saveButton = tkinter.Button(bottomFrame, text="Zapisz", command=saveToFile, width=18, height=1)

chartwstButton = tkinter.Button(chartFrame, text="Przebieg rejestracji", command=showChartwst, width=18, height=1)
timeButton = tkinter.Button(fFrame, text="Czas nadejścia fali N", command=showTime, width=18, height=1)
```

Realizowane jest to za pomocą funkcji *tkinter.Button*. Pierwszy jej argument informuje o wcześniej zdefiniowanej „ramce”, w której ma znajdować się przycisk. Zdefiniowanie „ramki” i jej położenia nastąpiło wcześniej. Kolejnym argumentem jest tekst jakim ma być podpisany przycisk, argumenty *width* i *height* określają szerokość i wysokość przycisku. Kluczowy jest argument *command*, do którego przypisujemy funkcję jaka ma być zrealizowana po naciśnięciu przycisku.


2.4. Podsumowanie rozdziału

Opracowany program można stosować do analizowania danych pomiarowych ze strzelań z dowolnej broni i różnych kalibrów użytej amunicji. Nie jest to w żaden sposób ograniczone i dla każdych danych otrzymywane są poprawne wyniki. Kod programu posiada prostą składnię, co zapewnia szybkie działanie programu. Umożliwia to analizowanie również bardzo dużych plików. Z doświadczeń autora wynika, że łączny czas wczytania pliku posiadającego kilkaset tysięcy rekordów

i szacowania parametrów fali N na komputerze z procesorem Intel Core i3 2,4 GHz nie przekracza kilku sekund.


Kod jest napisany tak, aby w łatwy sposób można było modyfikować go, zależnie od rodzaju analizowanego pliku. Systemy pomiarowe, które analizują i eksportują dane zarejestrowane przez czujniki akustyczne zapisują je czasem w różnej strukturze. Różnice wynikają najczęściej z rodzaju użytego spójnika, który rozdziela dane różnych kanałów. W programie wystarczy tylko zmienić spójnik w funkcji *openFile()*, w linijce z poleceniem *x.split()*.

```
for x in text:
    x = x.replace(',', '.')
    fields = x.split(';')
```





Jeśli spójnikiem oddzielającym część całkowitą od dziesiętnej nie jest przecinek, należy wpisać go w linijce *x.replace* (w przypadku, gdy jest to kropka, nie należy nic zmieniać, gdyż jest to poprawne zdefiniowanie liczby do dalszych obliczeń).

```
for x in text:
    x = x.replace('.', ',')
    fields = x.split(';')
```



W przypadku, gdy nasze analizowane pliki zawierają różną liczbę kanałów niż pliki wcześniejsze, efektywna modyfikacja polega na zastąpieniu liczby na nową w dwóch linijkach funkcji *openFile()*

```
column = [[] for i in range(4)]
for x in text:
    x = x.replace(',', '.')
    fields = x.split(';')
    for i in range(4):
        column[i].append(float(fields[i]))
```

Ostatnią istotną różnicą możliwą do powstania jest zdefiniowanie częstotliwości próbkowania. Napisany program czytuje podaną częstotliwość z pierwszego wiersza pliku, jak to zapisuje większość systemów. Niektóre systemy jednak nie podają jej gotowej, wtedy można ją obliczyć odwracając różnicę czasów rejestracji między sąsiednimi próbkami, deklarując to w linijce funkcji *openFile()*:

```
SampleFreq = float(text[0][17:-1].replace(',', '.'))
```

3. Testy walidacyjne programu

3.1. Wstęp do rozdziału

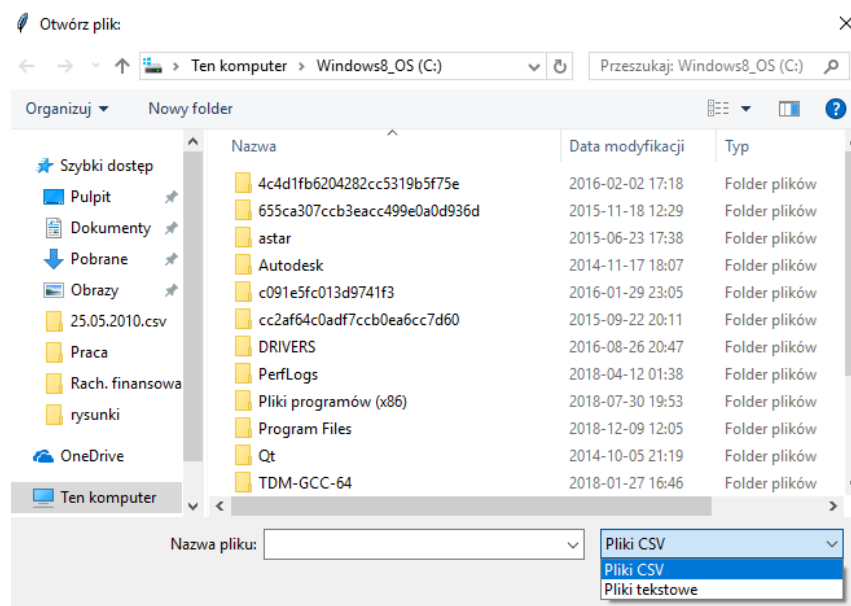
Walidacja w naukach technicznych i informatyce jest działaniem mającym na celu potwierdzenie w sposób udokumentowany i zgodny z założeniami, że procedury, procesy i czynności rzeczywiście prowadzą do zaplanowanych wyników. W przypadku niniejszej pracy pożądanym wynikiem działania programu jest poprawne oszacowanie parametrów fali N.

Walidację i testowanie działania programu podzielono na następujące części:

- sprawdzenie poprawności wczytania pliku,
- sprawdzenie poprawności obliczeń,
- sprawdzenie poprawności generowania wykresów,
- sprawdzenie poprawności zapisu oszacowanych parametrów do pliku.

3.2. Sprawdzenie poprawności wczytania pliku

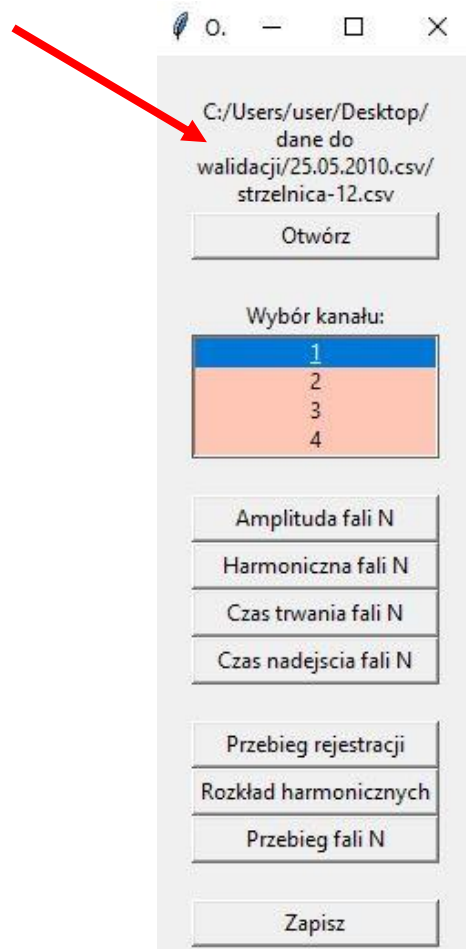
W celu otworzenia pliku należy nacisnąć przycisk *Otwórz* menu głównego programu. Po jego naciśnięciu otwiera się okno eksploratora:



Rysunek 16. Widok okna eksploratora umożliwiającego załadowanie pliku.

Pliki można wybierać spośród formatów .csv oraz .txt.

Po wybraniu, poprawność wczytania pliku potwierdzona jest wyświetleniem jego nazwy w górnej części menu głównego (ry.17).



Rysunek 17. Widok menu głównego programu z poprawnie wczytanym plikiem do analizy.

Plik został poprawnie wczytany. Wybranie interesującego kanału jest potwierdzone niebieskim zaznaczeniem.

3.3. Sprawdzenie poprawności obliczeń

W celu sprawdzenia poprawności szacowania parametrów fali N, przetestowano działanie programu na 34 plikach, pochodzących ze strzelań doświadczalnych przeprowadzonych przez pracowników z Zespołu Radioelektroniki WAT w dniu 25 maja 2010 roku na strzelnicy Wojskowej Akademii Technicznej. Uzyskane wyniki porównano z teoretycznymi wartościami. Strzelania doświadczalne przeprowadzono z użyciem pocisku 7,62x39 mm, gdzie jego długość wynosi 27 mm, a średnica 7,85 mm. Prędkość początkowa pocisku wynosi 715 m/s. Badania

przeprowadzono w temperaturze ok. 15°C, dla tej temperatury prędkość dźwięku wynosi ok. 340 m/s. Odległość d czujników od trajektorii lotu była w trakcie strzelań wielokrotnie zmieniana, jedną z nich jest odległość 15 m.

Na potrzeby walidacji założono, że jeśli dla odległości 15 m wyniki działania programu będą pokrywały się z obliczoną wartością teoretyczną, oznaczać to będzie poprawne działanie programu.

W przypadku pomiarów parametrów ośrodka parametrem znamionującym poprawność uzyskanych wyników jest czas propagacji fali N.

Korzystając ze wzoru (1) obliczono teoretyczny czas trwania fali N:

Liczba Macha: $715/340 = 2,1$.

$$T = \frac{1,82 \cdot 2,1 \cdot 0,00785 \cdot 3^{1/4}}{0,027^{1/4} \cdot 340 \cdot (2,1^2 - 1)^{3/8}} = \frac{0,0601}{218,324} = 0,0002752 \text{ [s]} = 0,275 \text{ ms}$$

Na rysunku 18 przedstawiono wycinek bazy otrzymanych wyników oszacowanych parametrów fali N.

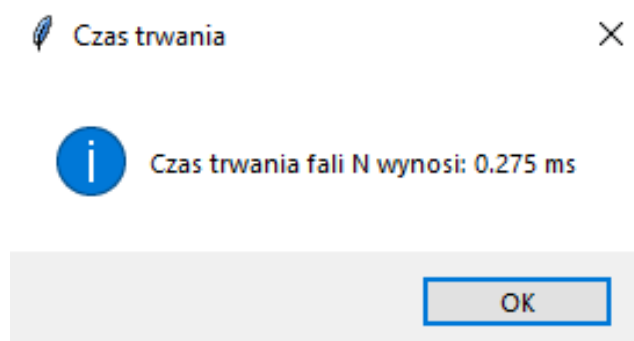
Plik	Kanał	Max	Czas trwania	F0	Czas rej.
C:/Users/user/Desktop/dane do walidacji/25.05.2010.csv/strzelnica-1.csv	kanał 1	179.321	0.29 ms	3276.8 Hz	2129.608 ms.
C:/Users/user/Desktop/dane do walidacji/25.05.2010.csv/strzelnica-11.csv	kanał 1	158.975	0.29 ms	3276.8 Hz	4707.626 ms.
C:/Users/user/Desktop/dane do walidacji/25.05.2010.csv/strzelnica-12.csv	kanał 1	221.182	0.275 ms	3640.89 Hz	2415.634 ms.
C:/Users/user/Desktop/dane do walidacji/25.05.2010.csv/strzelnica-13.csv	kanał 1	630.446	0.214 ms	4681.14 Hz	5904.266 ms.
C:/Users/user/Desktop/dane do walidacji/25.05.2010.csv/strzelnica-14.csv	kanał 2	563.605	0.214 ms	4681.14 Hz	9661.377 ms.
C:/Users/user/Desktop/dane do walidacji/25.05.2010.csv/strzelnica-15.csv	kanał 3	616.358	0.214 ms	4681.14 Hz	17949.738 ms.
C:/Users/user/Desktop/dane do walidacji/25.05.2010.csv/strzelnica-16.csv	kanał 4	533.734	0.214 ms	4681.14 Hz	14182.785 ms.
C:/Users/user/Desktop/dane do walidacji/25.05.2010.csv/strzelnica-17.csv	kanał 1	229.651	0.275 ms	3640.89 Hz	10676.788 ms.
C:/Users/user/Desktop/dane do walidacji/25.05.2010.csv/strzelnica-18.csv	kanał 2	204.436	0.259 ms	4096.0 Hz	8519.852 ms.
C:/Users/user/Desktop/dane do walidacji/25.05.2010.csv/strzelnica-100.csv	kanał 3	217.549	0.275 ms	3640.89 Hz	16970.52 ms.
C:/Users/user/Desktop/dane do walidacji/25.05.2010.csv/strzelnica-101.csv	kanał 4	490.713	0.214 ms	4681.14 Hz	8613.098 ms.
C:/Users/user/Desktop/dane do walidacji/25.05.2010.csv/strzelnica-102.csv	kanał 1	234.13	0.275 ms	3640.89 Hz	4876.175 ms.

Rysunek 18. Wycinek z bazy szacowanych parametrów fali N.

Oszacowana programowo wartość czasu trwania fali N w przypadku plików *strzelnica-12.csv* oraz *strzelnica-17.csv* pokrywa się z wyliczoną wartością teoretyczną. Pozostałe rozbieżności pomiędzy wartościami oszacowania za pomocą programu, a wartością teoretyczną mogą być związane z rozrzutem wyników strzelań wokół środka tarczy, oddalonej podczas badań o 15 m względem układu pomiarowego.

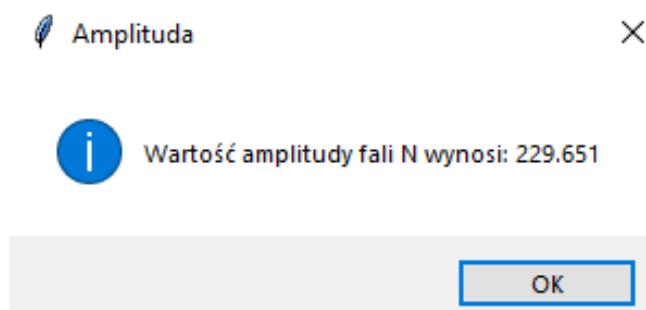
Z powyższego sprawdzenia wynika, że za pomocą opracowanego programu można w wiarygodny sposób szacować czas trwania fali N na podstawie danych zapisanych w analizowanych plikach pomiarowych.

Efekt działania programu po naciśnięciu przycisku *Czas trwania fali N* przedstawiono na rys.19.



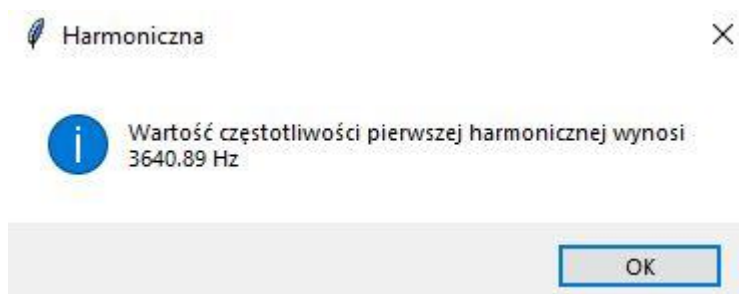
Rysunek 19. Okno dialogowe prezentujące działanie programu po naciśnięciu przycisku „Czas trwania fali N”.

Użycie przycisku *Amplituda fali N* powoduje wyświetlenie komunikatu z obliczoną wartością amplitudy analizowanej fali N (rys. 20).



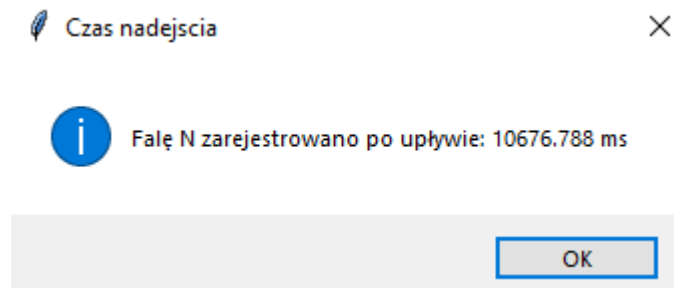
Rysunek 20. Okno dialogowe prezentujące działanie programu po naciśnięciu przycisku „Amplituda fali N”.

Wartość częstotliwości pierwszej harmonicznej fali N jest prezentowana w oknie dialogowym po naciśnięciu przycisku *Harmoniczna fali N* (rys. 21).



Rysunek 21. Okno dialogowe prezentujące działanie programu po naciśnięciu przycisku „Harmoniczna fali N”.

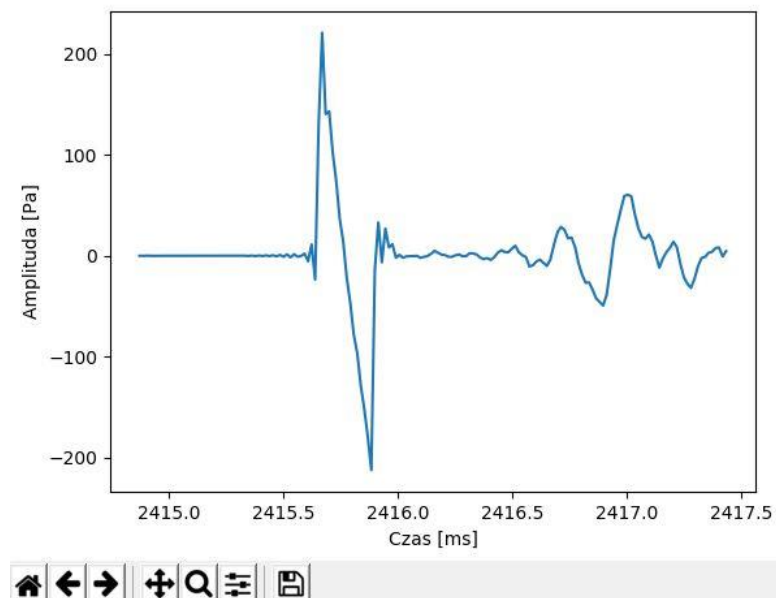
Naciśnięcie przycisku *Czas nadejścia fali N* powoduje wyświetlenie w oknie dialogowym komunikatu z obliczonym czasem nadejścia fali N do czujnika pomiarowego wybranego kanału.



Rysunek 22. Okno dialogowe prezentujące działanie programu po naciśnięciu przycisku „Czas nadejścia fali N”.

3.4. Sprawdzenie poprawności generowania wykresów

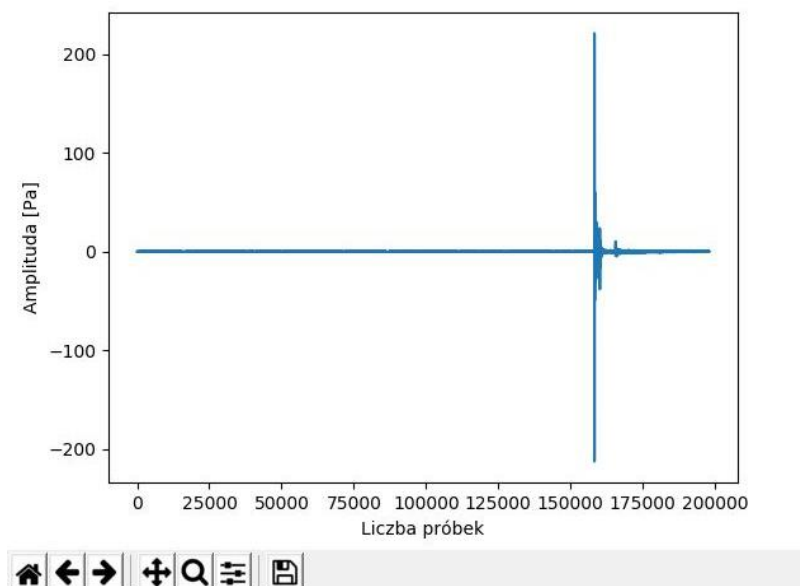
Wykres przebiegu fali N jest rysowany po naciśnięciu przycisku *Przebieg fali N*. Działanie programu przedstawiono na rys. 23.



Rysunek 23. Okno dialogowe prezentujące wykres przebiegu fali N.

Z wykresu można wywnioskować, że amplituda fali N ma wartość w przybliżeniu 220 Pa, a czas jej trwania wynosi około $0,2 \div 0,3$ ms. Informacje te pokrywają się z komunikatami programu z rys. 19 i 20, stąd wniosek, że generowany wykres jest prawidłowy.

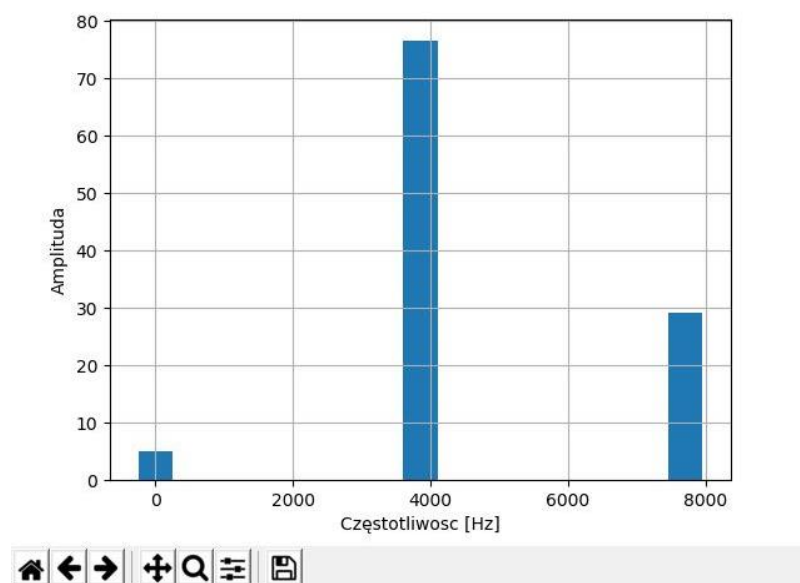
Po naciśnięciu przycisku *Przebieg rejestracji* generowany jest wykres przebiegu zapisanych danych znajdujących się w analizowanym pliku dla wybranego kanału (rys. 24).



Rysunek 24. Wykres przebiegu rejestracji danych w kanale pomiarowym.

Wykres wskazuje na obecność fali N o amplitudzie równej około 220 Pa, co w odniesieniu do wyników wcześniejszych sprawdzeń również wskazuje na poprawne działanie programu. Wygenerowane wykresy można dowolnie przybliżać, oddalać, wskazywać dokładne współrzędne i zapisywać w formacie *JPG* lub *PNG*.

Po naciśnięciu przycisku *Rozkład harmoniczných* generowany jest histogram rozkładu harmoniczných fali N (rys.25).

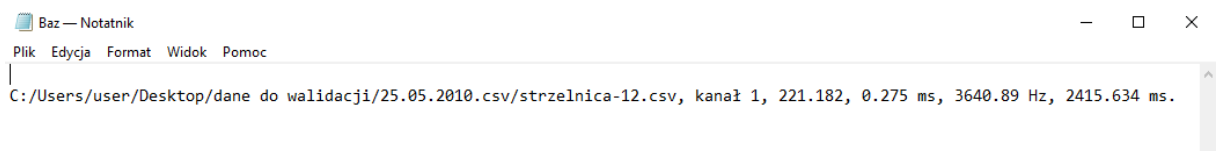


Rysunek 25. Rozkład harmoniczných fali N.

Wykres wskazuje na częstotliwość dominującą o wartości około 3600 Hz. Za pomocą narzędzia „Lupa” można udokładnić wskazanie wartości pierwszej harmonicznej fali N. W końcowym etapie informacja ta jest zgodna z komunikatem na rys. 21, wyświetlanym po naciśnięciu przycisku *Harmoniczna fali N*. Można zatem przyjąć, że rozkład wartości charakterystycznych częstotliwości harmonicznych fali N jest generowany prawidłowo.

3.5. Sprawdzenie poprawności zapisu oszacowanych parametrów do pliku

Przyciskiem *Zapisz* można wybrać plik, do którego pożądanego jest zapisanie oszacowanych parametrów fali N. Rezultat działania opcji *Zapisz w pliku* jest przedstawiony na rys. 26.



Rysunek 26. Plik z zapisanymi parametrami fali N.

Program poprawnie zapisuje oszacowane dane wraz z przypisaną nazwą pliku i wybranym kanałem, oddzielając dane przecinkami.

Podsumowanie

Cel mojej pracy został osiągnięty. Opracowałem program komputerowy do szacowania parametrów fali N na podstawie analizy zarejestrowanych danych w plikach pomiarowych pochodzących ze strzelań z broni palnej.

Wyniki przeprowadzonych testów walidacyjnych oprogramowania wskazują na jego prawidłowe działanie (oszacowane wartości czasu trwania fali N są zbieżne z obliczeniami teoretycznymi dla przyjętego modelu Whitmana).

Podzielenie programu na zbiór niezależnych i różnorodnych w działaniu funkcji, zwiększa niezawodność opracowanego oprogramowania oraz poprawia czytelność i łatwość wprowadzania modyfikacji kodu źródłowego.

Możliwość szybkiej prezentacji przebiegów czasowych i fali N, rozkładu jej widma oraz zobrazowanie całego zbioru wczytanych do analizy z pliku danych pomiarowych istotnie zwiększa funkcjonalność opracowanego programu. Program jest w całości aplikacją okienkową, co znacznie ułatwia i przyspiesza pracę. Właśnie opracowanie koncepcji i wykonanie interfejsu graficznego programu było dla mnie najbardziej pracochłonnym zadaniem. Każdy jego element został napisany przeze mnie samodzielnie bez wspomagania się gotowymi szablonami dostępnymi, które są dostępne np. w środowiskach *Matlab*[®] czy *Qt*[®].

Z perspektywy upływu czasu spędzonego nad opracowaniem programu, w dalszym ciągu widzę duże możliwości poprawy jego funkcjonalności w przyszłości. Z uwagi na ograniczony terminem realizacji pracy dyplomowej dalszy rozwój funkcjonalności oprogramowania nastąpi już w czasie studiów magisterskich.

Głównym zadaniem opracowanego programu jest poprawne szacowanie parametrów fali N. Realizacja tego warunku była dla mnie zajmującym zajęciem, szczególnie w przypadku opracowania algorytmu wyznaczania charakterystycznej częstotliwości pierwszej harmonicznej fali N.

W mojej ocenie opracowany program komputerowy do szacowania parametrów fali N na podstawie zarejestrowanych w plikach danych pomiarowych ze strzelań jest użytecznym narzędziem, gdyż pozwala na szybką analizę uzyskanych w czasie badań eksperymentalnych wyników. Perspektywa dalszego wykorzystania, uzyskanych przy pomocy opracowanego programu wyników, które można zapisać w wygodnym do dalszej obróbki formacie *.txt* stwarza interesujące możliwości.

Bibliografia

1. Rodzik D. (2017), *Pasywna lokacja akustyczna pocisków w ruchu naddźwiękowym*, rozdział w monografii pod red. naukową: Jerzy Lewitowicz, Krzysztof Dragan, Andrzej Leski, Sławomir Michalak, Mariusz Ziej, pt. „Problemy badań i eksploatacji techniki lotniczej”, Tom 9, ITWL, Warszawa, s. 73-84.
2. Maher C. (2007), *Acoustical Characterization of Gunshots*, 2007 IEEE Workshop on Signal Processing Applications for Public Security and Forensics., 109-114.
3. Thoma K., Hornemann U., Sauer M., Schneider E. (2005), *Shock waves – phenomenology, experimental and numerical simulation*. Meteoritics & Planetary Science 40, Nr 9/10, 1283–1298.
4. Maher C. (2006), *Modelling and signal processing of acoustic gunshot recordings*, IEEE Signal Processing Society 12th DSP Workshop, 10, 257-261.
5. Makinen T., Pertila P., Auranen P. (2009), *Supersonic bullet state estimation using particle filtering*, 2009 IEEE International Conference on Signal and Image Processing Applications, 150-155.
6. Maher C. (2018), *Principles of Forensic Audio Analysis*, Springer International Publishing.
7. Boschetti A., Massaron L. (2017), *Python. Podstawy nauki o danych*. Wydanie II, Helion.
8. Gągolewski M., Bartoszek M., Cena A. (2018), *Przetwarzanie i analiza danych w języku Python*, PWN.
9. Whitham G.B. (1956), *On the propagation of Weak Shock Waves*, Jour. Fluid Mech., 1, 290-318.
10. Whitham G.B. (1952), *The flow pattern of a supersonic projectile*, Commun. Pure and Appl. Math., 5, 301-348.
11. Rodzik D. (2011), *Investigation of N-Wave Propagation Parameters of Free Field Projectiles Moving*, Proceedings of 58th Open Seminar on Acoustics, T.2., Jurata, pp. 217-228.
12. EN ISO 17201-4 (2006), *Acoustics – Noise from shooting ranges – Part 4: Prediction of projectile sound*.
13. Snow W.B. (1967), *Survey of acoustic characteristics of bullet shock waves*, IEEE Transactions on Audio and Electroacoustics, Au-15, 4, 161-176.

Załącznik 1. Pełny kod programu

```

1 import tkinter
2 import sys
3 import numpy as np
4 import matplotlib.pyplot as plt
5 from tkinter import filedialog
6 from tkinter import messagebox
7
8 fileLoaded = False
9 selection = None
10
11 def fileNotLoadedException():
12     global fileLoaded
13     if not fileLoaded:
14         tkinter.messagebox.showwarning("Nieprawidłowe polecenie",
15                                         "Najpierw wybierz plik!")
16         return True
17     return False
18
19 def noSelectionException():
20     global selection
21     if not selection:
22         tkinter.messagebox.showwarning("Nieprawidłowe polecenie",
23                                         "Najpierw wybierz kanał!")
24         return True
25     return False
26
27 def findBegin(channel, index):
28     while index >= 0 and channel[index] > channel[index-1]:
29         index -= 1
30     return index
31
32 def findEnd(channel, index):
33     while index < len(channel) and channel[index] <= 0:
34         index += 1
35     return index
36
37 def openFile():
38     global filenameopen
39     global text
40     global SampleFreq
41     global column
42     global fileLoaded
43     global selectionLabel
44
45     filenameopen = tkinter.filedialog.askopenfilename(initialdir = "/",
46                                                         title = "Otwórz plik:", filetypes = (("Pliki CSV", "*.csv"),
47                                                         ("Pliki tekstowe", "*.txt")))
48
49     if not filenameopen:
50         tkinter.messagebox.showwarning("Niepowodzenie", "Nie wybrano pliku!")
51         return
52
53     plik = open(filenameopen)
54     text = plik.readlines()
55     plik.close()

```

```

52 SampleFreq = float(text[0][17:-1].replace(',', '.'))
53
54 text = text[13:]
55 column = [[] for i in range(4)]
56
57 for x in text:
58
59     x = x.replace(',', '.')
60     fields = x.split(' ')
61     for i in range(4):
62         column[i].append(float(fields[i]))
63
64 fileLoaded = True
65 selectionLabel.config(text=filenameopen)
66
67
68 def onselect(event):
69     global selection
70     widget = event.widget
71     selection = widget.get(int(widget.curselection()[0]))
72
73 def showMax():
74     global selection
75     global column
76     if fileNotLoadedException():
77         return
78     if noSelectionException():
79         return
80     tkinter.messagebox.showinfo("Amplituda", "Wartość amplitudy fali N wynosi:
81     " + str(max(column[int(selection)-1])))
82
83 def showTime():
84     global selection
85     global column
86     global SampleFreq
87     if fileNotLoadedException():
88         return
89     if noSelectionException():
90         return
91
92     index = int(selection) - 1
93     beg = findBegin(column[index], column[index].index(max(column[index])))
94     time=1/SampleFreq*beg
95     time=time*1000
96     time=round(time,3)
97     tkinter.messagebox.showinfo("Czas nadejścia",
98     "Fale N zarejestrowano po upływie: " + str(time) + " ms")

```

```

98
99
100 def showFreq():
101     global SampleFreq
102     global column
103     global selection
104     global czestotliwosc
105     if fileNotLoadedException():
106         return
107     if noSelectionException():
108         return
109
110     index = int(selection) - 1
111     beg, end = findBegin(column[index], column[index].index(max(column[index]))), findEnd(column[index], column[index].index(min(column[index])))
112     fu1 = np.fft.fft(column[index][beg:end])
113     n1=round((len(fu1))/2)
114     fu2=abs(fu1[:n1])
115     m=1
116     maxii=0
117     while m<n1:
118         if fu2[m]>maxii:
119             maxii=fu2[m]
120             maxi2=m
121         m=m+1
122     czestotliwosc=maxi2*(SampleFreq/(2*n1))
123     czestotliwosc=round(czestotliwosc,2)
124
125
126
127     tkinter.messagebox.showinfo("F0", "Wartość częstotliwości pierwszej
128         harmonicznej wynosi " + str(czestotliwosc) + " Hz")
129
130 def showPeriod():
131     global SampleFreq
132     global selection
133     global column
134     if fileNotLoadedException():
135         return
136     if noSelectionException():
137         return
138     index = int(selection)-1
139     period = 1 / SampleFreq * (findEnd(column[index],
140         column[index].index(min(column[index]))))
141     - findBegin(column[index], column[index].index(max(column[index]))))
142     period = period*1000
143     period = round(period, 3)
144     tkinter.messagebox.showinfo("Czas trwania", "Czas trwania fali N wynosi:
145         " + str(period) + " ms")
146
147 def showChartwst():
148     global selection
149     global column
150     global SampleFreq
151     if fileNotLoadedException():
152         return
153     if noSelectionException():
154         return
155     index = int(selection)-1

```

```

155     y= column[index]
156
157
158     plt.ylabel("Amplitude")
159     plt.xlabel("Liczba próbek")
160
161     plt.plot(y)
162     plt.show()
163
164 def showChart():
165     global selection
166     global column
167     global SampleFreq
168     if fileNotLoadedException():
169         return
170     if noSelectionException():
171         return
172     index = int(selection)-1
173     beg, end = findBegin(column[index], column[index].index(max(column[index])))
174     ), findEnd(column[index], column[index].index(min(column[index])))
175
176
177     y= column[index][beg-50:end+100]
178     z=len(y)
179
180     x=np.linspace(1000*(1/SampleFreq)*(beg-50), 1000*(1/SampleFreq)*(end+100),
181                  z)
182     plt.ylabel("Amplitude")
183     plt.xlabel("Time [ms]")
184
185     plt.plot(x,y)
186     plt.show()
187 def saveToFile():
188     global column
189     global selection
190     global SampleFreq
191     global czestotliwosc
192     global filenameopen
193     if fileNotLoadedException():
194         return
195     if noSelectionException():
196         return
197
198     index = int(selection) - 1
199
200     maxi = str(max(column[index]))
201
202     period = 1 / SampleFreq * (findEnd(column[index], column[index].
203                                     index(min(column[index])))
204     - findBegin(column[index], column[index].index(max(column[index]))))
205     period=period*1000
206     period=round(period,3)
207
208     beg, end = findBegin(column[index], column[index].index(max(column[index]
209     )), findEnd(column[index], column[index].index(min(column[index])))
210     fu1 = np.fft.fft(column[index][beg:end])
211     n1=round((len(fu1))/2)
212     fu2=abs(fu1[:n1])

```

```

212 while m<n1:
213     if fu2[m]>maxii:
214         maxii=fu2[m]
215         maxi2=m
216     m=m+1
217 czestotliwosc=maxi2*(SampleFreq/(2*n1))
218 czestotliwosc=round(czestotliwosc,2)
219
220 beg = findBegin(column[index], column[index].index(max(column[index])))
221 time=1/SampleFreq*beg
222 time=time*1000
223 time=round(time,3)
224
225
226 filename = tkinter.filedialog.asksaveasfilename(initialdir="/", title=
227     "Zapisz plik:",
228     filetypes=(("Pliki CSV", "*.csv"), ("Pliki tekstowe", "*.txt")))
229 if not filename:
230     tkinter.messagebox.showwarning("Niepowodzenie", "Nie wybrano pliku!")
231     return
232
233
234 plik = open(filename,'a')
235 plik.writelines('\n' + str(filenameopen) + ', ' + 'kanał ' + str(selection) +
236     ', ' + str(maxi) + ', ' + str(period) + ' ms, '
237     + str(czestotliwosc) + ' Hz, ' + str(time) + ' ms.')
238 plik.close()
239
240 def showFFT():
241     global SampleFreq
242     global column
243     global selection
244     if fileNotLoadedException():
245         return
246     if noSelectionException():
247         return
248
249     index = int(selection) - 1
250     beg, end = findBegin(column[index], column[index].index(max(column[index])))
251     , findEnd(column[index], column[index].index(min(column[index])))
252     fu = np.fft.fft(column[index][beg:end])
253     n=len(fu)
254     f=np.linspace(0, SampleFreq, n)
255     plt.bar(f[:n // 6], np.abs(fu)[:n // 6] * 1 / n, width=500)
256     plt.ylabel("Amplitude")
257     plt.xlabel("Frequency [Hz]")
258     plt.grid()
259     plt.show()
260
261 app = tkinter.Tk()
262 app.title("Odczyt danych fali N")
263 content = tkinter.Frame(app)
264 topFrame = tkinter.Frame(content)
265 channelsFrame = tkinter.Frame(content)
266 fFrame = tkinter.Frame(content)
267 chartFrame = tkinter.Frame(content)
268 bottomFrame = tkinter.Frame(content)
269
270 selectionLabel = tkinter.Label(topFrame, text="<brak otwartego pliku>",
271     wraplength=130)

```



```

267 openButton = tkinter.Button(topFrame, text="Otwórz", command=openFile, width=1
268
269 channelsLabel = tkinter.Label(channelsFrame, text="Wybór kanału:", width=22,
270                               height=1)
271 channelsList = tkinter.Listbox(channelsFrame, selectmode="SINGLE", justify=
272                               "center", bg="#FFC6B5", width=22, height=4)
273 channelsList.bind('<<ListboxSelect>>', onselect)
274 channelsList.insert(1, "1")
275 channelsList.insert(2, "2")
276 channelsList.insert(3, "3")
277 channelsList.insert(4, "4")
278
279 maxButton = tkinter.Button(fFrame, text="Amplituda fali N", command=showMax,
280                             width=18, height=1)
281 fButton = tkinter.Button(fFrame, text="Harmoniczna fali N", command=showFreq,
282                             width=18, height=1)
283 periodButton = tkinter.Button(fFrame, text="Czas trwania fali N",
284                                 command=showPeriod, width=18, height=1)
285
286 chartButton = tkinter.Button(chartFrame, text="Przebieg fali N",
287                                 command=showChart, width=18, height=1)
288 fftButton = tkinter.Button(chartFrame, text="Rozkład harmonicznych",
289                                 command=showFFT, width=18, height=1)
290
291
292 saveButton = tkinter.Button(bottomFrame, text="Zapisz", command=saveToFile,
293                             width=18, height=1)
294
295 chartwstButton = tkinter.Button(chartFrame, text="Przebieg rejestracji",
296                                 command=showChartwst, width=18, height=1)
297 timeButton = tkinter.Button(fFrame, text="Czas nadejścia fali N",
298                             command=showTime, width=18, height=1)
299
300 content.grid(column=0, row=0, padx=10, pady=10)
301
302 topFrame.grid(column=1, row=0, pady=10)
303 selectionLabel.grid(column=1, row=0)
304 openButton.grid(column=1, row=1)
305
306 channelsFrame.grid(column=1, row=2, pady=10)
307 channelsLabel.grid(column=1, row=2)
308 channelsList.grid(column=1, row=3)
309
310 fFrame.grid(column=1, row=4, pady=10)
311 maxButton.grid(column=1, row=4)
312 fButton.grid(column=1, row=5)
313 periodButton.grid(column=1, row=6)
314
315 chartFrame.grid(column=1, row=8, pady=10)
316 chartButton.grid(column=1, row=12)
317 fftButton.grid(column=1, row=9)
318
319 bottomFrame.grid(column=1, row=10, pady=10)
320 saveButton.grid(column=1, row=10)
321
322 chartwstButton.grid(column=1, row=8)
323 timeButton.grid(column=1, row=7)
324
325 app.mainloop()

```

(strona celowo zostawiona pusta)

Warszawa, 201... r.

Wojskowa Akademia Techniczna

Wydział Mechatroniki i Lotnictwa

Imię i nazwisko studenta: **plut. pchor. Dawid BREJECKI**Numer albumu: **62405**

OŚWIADCZENIE

Świadomy(a) odpowiedzialności karnej z tytułu naruszenia przepisów ustawy o prawie autorskim i prawach pokrewnych (Dz. U. nr 80 poz. 904 z 2000 r. ze zmianami) oraz konsekwencji dyscyplinarnych określonych w ustawie z dnia 27 lipca 2005 r. prawo o szkolnictwie wyższym (Dz.U. Nr 164 poz. 1365, z późn. zm.)¹⁾, a także odpowiedzialności cywilnoprawnej oświadczam, że przedkładana praca dyplomowa pt. **„Program komputerowy do wyznaczania parametrów fali N na podstawie zarejestrowanych danych ze strzelań”** została napisana przeze mnie samodzielnie i nie była wcześniej podstawą żadnej innej urzędowej procedury oświadczam, że wyżej wymieniona praca dyplomowa nie narusza praw autorskich w rozumieniu ustawy o prawie autorskim i prawach pokrewnych innych osób oraz dóbr osobistych chronionych prawem cywilnym. Wszystkie informacje umieszczone w pracy, uzyskane ze źródeł pisanych i elektronicznych oraz inne informacje, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami.

.....
/data, podpis studenta/

¹⁾ Ustawa z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym:

Art. 214 ust. 4. „W razie podejrzenia popełnienia przez studenta czynu polegającego na przypisaniu sobie autorstwa istotnego fragmentu lub innych elementów cudzego utworu rektor niezwłocznie poleca przeprowadzenie postępowania wyjaśniającego.”;

Art. 214 ust. 6. „Jeżeli w wyniku postępowania wyjaśniającego zebrany materiał potwierdza popełnienie czynu, o którym mowa w ust. 4, rektor wstrzymuje postępowanie o nadanie tytułu zawodowego do czasu wydania orzeczenia przez komisję dyscyplinarną oraz składa zawiadomienie o popełnieniu przestępstwa.”.

Wyrażam zgodę na udostępnienie mojej pracy dyplomowej przez Archiwum WAT w czytelni i w ramach wypożyczeń międzybibliotecznych.

.....
/data, podpis studenta/