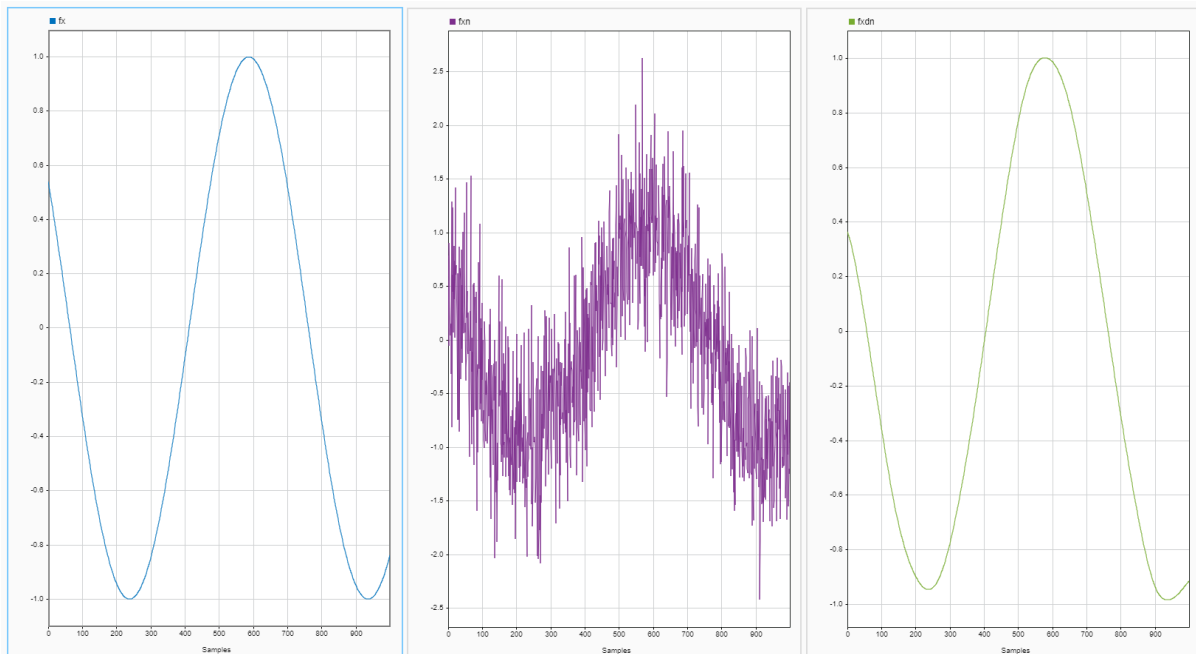


POMIAR I APROKSYMACJA RUCHU  
UPROSZCZONEGO WAHADŁA  
MATEMATYCZNEGO

## 1. Wstęp teoretyczny oraz cel projektu

Idea naszego projektu narodziła się podczas sprawdzania, jak zaimplementowana w pakiecie MATLABa (Wavelet Toolbox) funkcja odszumiająca przy pomocy falek `wdenoise` radzi sobie z wygenerowanym sztucznie białym szumem Gaussowskim. Rezultaty były zaskakująco dobre - nawet przy bardzo widocznym szumie otrzymywaliśmy sygnał zbliżony do pierwotnego, pozbawionego szumów.



od lewej kolejno czysty sygnał pierwotny, sygnał zaszumiony oraz sygnał odszumiony przy pomocy falki biortogonalnej

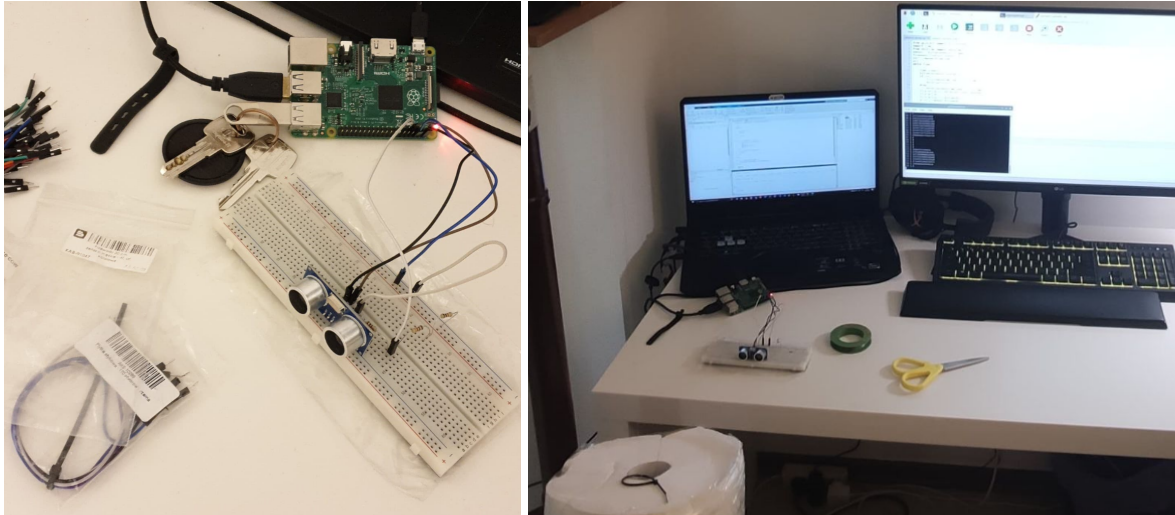
Postanowiliśmy zatem sprawdzić, czy efektywność odszumiania będzie równie wysoka w przypadku bardziej naturalnych szumów, powstałych podczas szczegółowo opisanego poniżej empirycznego zbierania danych. Przeprowadziliśmy doświadczenie polegające na wyznaczeniu przyspieszenia ziemskiego za pomocą wahadła matematycznego. Zebraliśmy łącznie kilkanaście zróżnicowanych pomiarów - sygnałów o charakterze sinusoidalnym reprezentujących odległość, które były w różny sposób zniekształcone/zaszumione (z uwagi chociażby na uproszczenia podczas realizacji doświadczenia, choć niektóre szумы były umyślne - w celu symulacji potencjalnych zakłóceń podczas tego typu pomiarów), co widocznie zagłuszało niesioną przez nie informację. Napisaliśmy również prostą funkcję umożliwiającą obliczenie przybliżonej wartości przyspieszenia na podstawie pomiarów, lecz z uwagi na jej uproszczoną implementację, w celu uzyskania miarodajnych wyników, wymagała ona względnie "gładkich" sygnałów. Postanowiliśmy zatem przetworzyć otrzymane sygnały, starając się zredukować zniekształcenia oraz częściowo pozbywając się szumów.

Otrzymane przez nas rezultaty były z wielu względów niesatysfakcjonujące - sygnały odbiegały mocno od ideału, nadal były "poszarpane" - ich wartość informatywna niekoniecznie wzrosła, co uniemożliwiało nam skuteczne obliczenie wartości przyspieszenia.

Postawiliśmy sobie dodatkowy cel - zaczęliśmy się zastanawiać, czy możemy w jakiś sposób zoptymalizować przetwarzanie sygnałów i po chwili wpadliśmy na pomysł użycia transformaty Fouriera - szczególnie z uwagi na charakterystykę przetwarzanych sygnałów zbliżoną do sygnałów z bazy. W tym celu napisaliśmy funkcję, opisaną szczegółowo poniżej, która przy pomocy wspomnianej transformaty, odszumiała zebrane przez nas sygnały. Otrzymaliśmy dzięki temu bardzo ciekawe wyniki, zaprezentowane w dalszej części raportu.

## 2. Pozyskiwanie i dobór danych

W celu pozyskania empirycznych danych postanowiliśmy wykorzystać dostępne nam narzędzia, aby w jak najbardziej wiarygodny sposób przedstawić rzeczywisty ruch wahadła matematycznego cyfrowo. Postępujemy się mikrokomputerem Raspberry Pi oraz dedykowanym mu czujnikiem odległości działającym na zasadzie emisji ultradźwięków.



Do obsługi układu wykorzystaliśmy bibliotekę GPIO Zero umożliwiającą łatwy odczyt informacji zbieranej przez czujnik. Szybko zorientowaliśmy się jednak, że proponowane w niej klasy nie pozwalają rozwinąć czujnikowi skrzydeł i bardzo ograniczają częstotliwość próbkowania. Postanowiliśmy napisać na własną rękę program wyliczający odległość od wahadła na podstawie różnicy czasu pomiędzy emisją i odczytem sygnału dźwiękowego generowanego przez czujnik. Wykorzystany kod zostanie załączony wraz z raportem.

W celu pozyskania sinusoidalnego sygnału skonstruowaliśmy uproszczone wahadło składające się z cienkiej żyłki, ciężkiej rolki papieru i wysokiego wieszaka. Choć konstrukcja była prowizoryczna, to otrzymane wyniki były wystarczająco dokładne, aby móc je porównywać do sztucznie wygenerowanego, teoretycznego wzorca opisanego równaniem matematycznym.

Program pozwolił nam na regulację częstotliwości próbkowania, co pozwoliło nam później sprawdzić w jak bardzo istotny sposób zmiany zagęszczenia pomiarów w jednostce czasu wpływają na wartość informacyjną serii danych. Staraliśmy się również wygenerować różne rodzaje zakłóceń i szumów, które moglibyśmy w dalszej części zniwelować testowanymi przez nas narzędziami. Zderzenie z rzeczywistością zupełnie odmieniło nasze podejście do dalszej części doświadczenia, gdyż zdecydowana większość zakłóceń nie dała się w sensowny sposób opanować znanymi nam algorytmami. Przytrzymanie wahadła, jego nierównomierne rozbijanie, czy też chwilowe jego przystąpienia kompletnie zmieniały charakter ruchu i choć moglibyśmy łatwo wyekstrahować sam ruch wahadła i oczyścić go z wyżej wymienionych szumów, to powstałe zmiany uniemożliwiłyby realizację celów naszego projektu, albowiem potrzebowaliśmy jak najdokładniej obliczyć okres ruchu, a taka wysokopoziomowa ekstrakcja zaburzyłaby go znacząco. Zakładaliśmy badanie algorytmów działających raczej lokalnie i wyznaczenie przyspieszenia ziemskiego, zatem jedyny szum o odpowiadającej naszym oczekiwaniom częstotliwości oddziaływania udało nam się wygenerować poprzez niewielkie drgania czujnikiem podczas pomiaru. Wszelkie otrzymane przez nas dane załączone zostały wraz z raportem, jednak w naszych obliczeniach i analizach zajmowaliśmy się jedynie najlepszymi pomiarami z szumem w postaci drgań oraz seriami w żaden sposób

intencjonalnie przez nas niezakłóconymi, i tak już obarczonymi błędami i niedokładnością naszych pomiarów.

Sam odczyt i zapis danych przeprowadziliśmy na dwa sposoby. Pierwszy z nich był realizowany z wykorzystaniem klas gotowej biblioteki, często powtarzając te same wartości pomiarów, co z pewnością wpływa na wartość informacyjną danych, drugi natomiast, wykorzystywał gotowe narzędzia tylko w połowie i pozwolił na znacznie szybsze próbkowanie, ale zdarzało mu się, z nieznanymi nam przyczynami, generować wadliwe odczyty o niezwykle dużej wartości. Błędy znaczące zmuszeni byliśmy usunąć skonstruowanym przez nas algorytmem do oczyszczania niewiarygodnych odczytów. Pomimo wysokiej dokładności wykorzystywanego czujnika, jego oprogramowanie pozostawiało wiele do życzenia, co sprawiało, że zachowywał się on niedeterministycznie, podobnie jak wahadło przez nas skonstruowane. Z początkowego planu porównywania pomiarów z wygenerowanymi przez nas zakłóceniami i bez, zdecydowaliśmy się naprawiać cyfrowo zapisany przez nas ruch i sprawdzać jak poszczególne metody przybliżają go do teoretycznego, fizycznego ideału ruchu wahadłowego.

Jednym z nasuwających się na myśl sposobów redukcji błędów jest uśrednienie otrzymanych serii danych, jednak zdecydowaliśmy się działać na pojedynczych próbach z dwóch powodów. Po pierwsze, naszym problemem nie była dokładność pomiaru, bo czujnik był bardzo precyzyjny, ale losowe, silne zaburzenia, co w przypadku uśredniania mogłoby skutkować zniekształceniem kilku bardzo dokładnych pomiarów jednym, wysoce niedokładnym. Po drugie, program napotykał zacięcia przy wysokim próbkowaniu spowodowane koniecznością szybkiego zapisu dużej ilości informacji do pliku, co sprawiło, że nasze punkty pomiarowe były bardzo rozrzucone w czasie i mogłyby się uśredniać w złych momentach ruchu.

### 3. Narzędzia i źródła wiedzy

Napisaliśmy w MATLABie funkcję **przysp**, obliczającą przyspieszenie ziemskie na podstawie ruchu wahadła matematycznego, zapisanego w postaci danych odległości od czujnika w zależności od czasu pomiaru. Algorytm porównując ze sobą dwa kolejne pomiary ustala, czy wykres w danym miejscu rośnie, czy maleje i zestawia to ze stanem poprzedniego porównania. Jeśli stan się zmienia z "rosnący" na "malejący" - znalezione zostało maksimum lokalne - punkt ten zostaje zapisany jako granica między kolejnymi okresami. Na podstawie tych punktów wyliczony jest średni okres dla danej serii danych.

Ze względu na charakter działania algorytmu, nie sprawdza się on przy zaszumionych sygnałach, ponieważ znajduje granice między okresami (maksima) w miejscach wystąpień szumów i "peaków", tym samym zaburzając uśredniony okres i wpływając na wynik działania całej funkcji.

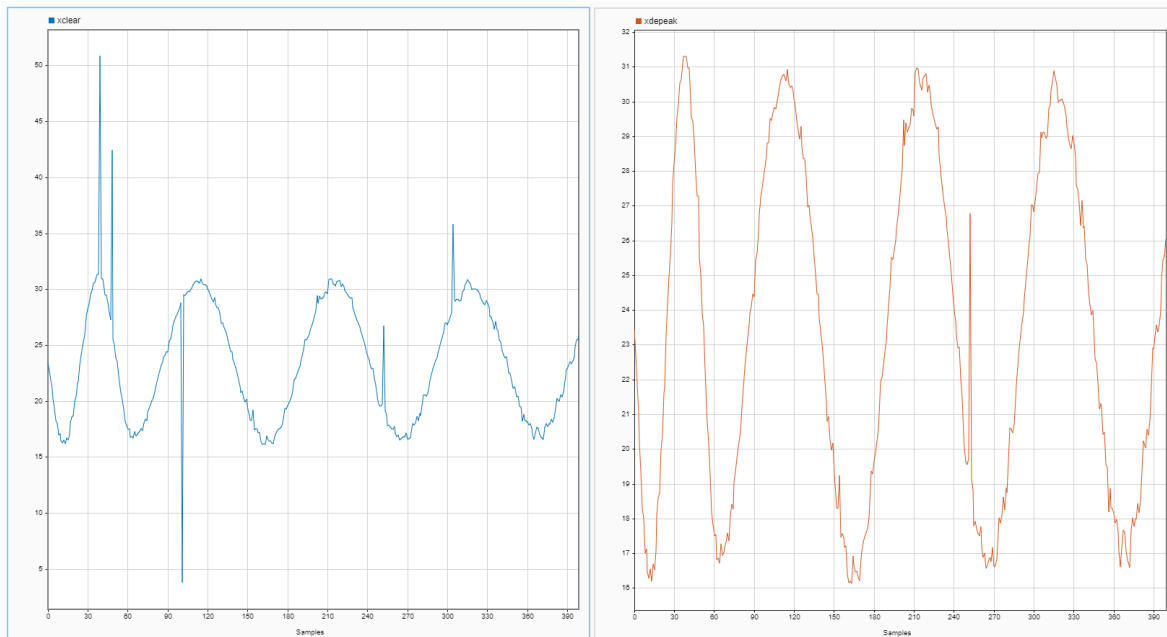
Na podstawie tak obliczonego okresu i wiedzy o długości wahadła  $l = 1,05$  m, obliczone zostaje przyspieszenie ziemskie z przekształconego wzoru na okres drgań wahadła matematycznego

$$g = \frac{4\pi^2 \cdot l}{T^2}, \text{ gdzie } T - \text{średni okres drgań obliczony na podstawie danego sygnału.}$$

Po wstępnym przeanalizowaniu wykresów zależności odległości od czasu, stwierdziliśmy, że do ciekawych wniosków doprowadzić nas może wyliczona z danych prędkość. Z tego powodu napisaliśmy w MATLABie funkcję **predkosc**, która dla każdej pary sąsiadujących pomiarów liczy średnią prędkość wahadła między tymi pomiarami.

Z uwagi na powstałe podczas niektórych pomiarów nienaturalne zniekształcenia sygnału w formie peaków (spowodowane prawdopodobnie błędnym działaniem czujnika), naturalnym pierwszym krokiem oczyszczania sygnału, wydało nam się pozbycie się ich. Napisaliśmy w tym celu funkcję

**depeak** w MATLABie, która przy użyciu progowania względem obliczanych w pewien sposób (z uwzględnieniem mediany) współczynników zastępuje fragmenty z nienaturalnymi peakami wartością poprzedniej próbki.



przykładowe oczyszczanie sygnału z nienaturalnych peaków

Jak widzimy, efekty działania funkcji są zadowalające, pozostał jeden peak, który z uwagi na umiejscowienie w sygnale i jego otoczenie, wymagałby bardziej lokalnego podejścia do problemu.

Na tak oczyszczonym sygnale pracowaliśmy przy użyciu wbudowanej w paczkę Wavelet Toolbox funkcji **wdenoise**, której zadaniem było oczyszczanie sygnałów z szumów. W celu uzyskania najlepszego odszumiania, jako parametr **wmaxlev** określający maksymalny poziom dekompozycji falkowej podawaliśmy maksymalną możliwą wartość, tzn.  $\text{floor}(\log_2 N)$ , gdzie  $N$  - liczba próbek w sygnale. Testowaliśmy falki z różnych rodzin, m.in. falkę Haara, Daubechies, biortogonalną, Coiflet oraz Symlet, kolejno opisane w pomiarach jako **xhaar**, **xdb**, **xbior**, **xcoif** oraz **xsym**. Z każdej rodziny dobraliśmy falkę możliwie zbliżoną charakterystyką do badanego przez nas problemu.

Po niezbyt satysfakcjonujących wynikach testów przeprowadzonych przy użyciu wyżej opisanej funkcji, postanowiliśmy skupić się na transformacie Fouriera. W tym celu napisaliśmy w MATLABie funkcję **FFT**, która przy pomocy transformaty Fouriera, a także progowania względem progu ustalanego przez użytkownika w argumencie funkcji oczyszcza sygnał z potencjalnych szumów. Funkcja oprócz odszumionego sygnału zwraca również liczbę niezerowych składowych, z których "zbudowany" został nowy sygnał. Jednym z wygenerowanych przez nas sygnałów jest **xfourierpower**, dla każdego pomiaru był on tworzony przy ściśle określonym progu (200), względem którego ma działać funkcja.

Oprócz wspomnianego powyżej sygnału, chcieliśmy również sprawdzić, jak bliski ideałowi sygnał możemy uzyskać przy użyciu transformaty Fouriera. W tym celu napisaliśmy skrypt o nazwie **bestFFT**, którego działanie polegało na odszumianiu sygnału **xdepeak** dla kolejnych wartości progu (z przedziału 1-1000) oraz obliczaniu błędu średniokwadratowego względem wygenerowanego ideału. Końcowo otrzymaliśmy odszumiony sygnał o możliwie minimalnej wartości MSE - **xfourier**.

Z uwagi na niedokładność naszych pomiarów i z faktu, że nie udało nam się wygenerować idealnego sygnału pomiarami, postanowiliśmy wygenerować idealny sygnał za pomocą wzorów na sinusoidę.

Napisałiśmy funkcję **ideal**, w której manipulujemy wykresem sinusoidy za pomocą okresu obliczonego w funkcji przysp dla xfourierpower (dla tej aproksymacji funkcja przysp wyliczała nam najbardziej zbliżony okres do oryginału) oraz amplitudy drgań.

W naszych pomiarach mierzyliśmy wyżej wspomniany błąd średniokwadratowy, a także entropię sygnałów. Wartości otrzymaliśmy przy użyciu kolejno funkcji **measerr** oraz **entropy** (ponieważ docelowo funkcja ta mierzy entropię obrazów w skali szarości, w celu otrzymania miarodajnego wyniku znormalizowaliśmy wejściowy wektor, dzieląc go przez maksymalną wartość modułu spośród jego składowych).

Wiedzę niezbędną nam do realizacji projektu pozyskaliśmy na wykładzie oraz uzupełniliśmy, korzystając głównie z dokumentacji znalezionej na stronie mathworks, m.in. [Wavelet signal denoising](#) czy też [Introduction to Wavelet Families](#). Wspieraliśmy się również znaleźćm na portalu YouTube filmem pt. [Denoising Data with FFT \[Matlab\]](#).

#### 4. Zestawienie wyników

Przykładowo przedstawimy poniżej cztery pomiary zróżnicowane pod względem typu zakłócenia lub próbkowania. Dla każdego z tych sygnałów wykreowaliśmy sygnał idealny, względem którego porównywaliśmy wartości błędu średniokwadratowego oraz entropii.

pomiarladny1_wahadlo_0001				
	MSE	entropia	g	l. składowych (dla Fouriera)
depeak	0,4198	6,7528	822,1319	x
haar	0,5100	6,3645	17,7230	x
db	0,3971	6,7962	134,0305	x
bior	0,4022	6,7508	159,5073	x
coif	0,3942	6,8316	111,4969	x
sym	0,3993	6,7532	217,1072	x
fourier	0,4004	6,7888	9,9049	33
fourierpower	2,5338	6,8721	9,9049	11
ideal	x	6,7702	x	x

pomiarldrgan3_wahadlo_0001				
	MSE	entropia	g	l. składowych (dla Fouriera)
depeak	1,1664	6,5457	3598,7100	x
haar	1,1684	6,1991	311,9615	x
db	1,1082	6,6106	1177,2417	x
bior	1,1041	6,5272	648,3568	x
coif	1,0966	6,5521	701,2627	x
sym	1,1048	6,5680	756,2433	x
fourier	1,1700	6,5791	448,3568	33
fourierpower	3,0803	6,6154	10,2561	7
ideal	x	6,3289	x	x

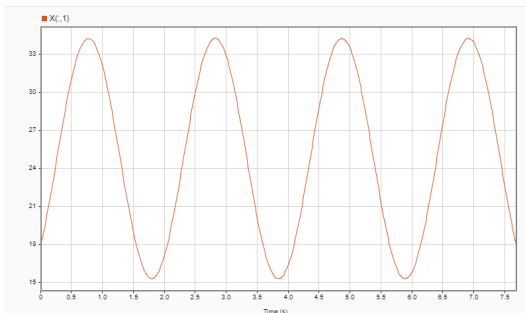
pomiar6_wahadlo_01				
	MSE	entropia	g	l. składowych (dla Fouriera)
depeak	11,6653	5,5522	9,9045	x
haar	12,6481	5,0200	10,3568	x
db	11,6299	5,5782	10,2014	x
bior	11,6404	5,4892	10,2014	x
coif	11,6050	5,7733	10,2014	x
sym	11,6487	5,5944	10,2014	x
fourier	11,6642	5,5326	10,2014	17
fourierpower	13,7917	5,8154	10,8371	5
ideal	x	5,7213	x	x

pomiarldrganie9_wahadlo_0001				
	MSE	entropia	g	l. składowych (dla Fouriera)
depeak	2,3173	6,2929	3183,6456	x
haar	2,3621	5,6896	112,0031	x
db	2,2957	6,4437	451,6361	x
bior	2,2994	6,3082	833,6443	x
coif	2,2969	6,3703	833,6443	x
sym	2,2982	6,3233	718,8055	x
fourier	1,9438	6,5544	11,2372	3
fourierpower	1,9438	6,5544	11,2372	3
ideal	x	6,2538	x	x

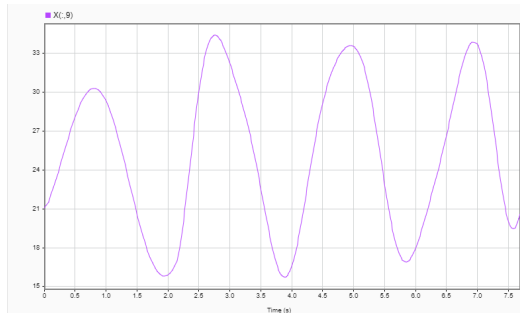
Jak możemy zauważyć w tabelach wyżej, metoda fourierpower pod względem błędu średniokwadratowego okazała się najmniej dokładna, co nie jest zaskakujące, ponieważ jak widzimy do aproksymacji została użyta bardzo mała ilość składowych. Patrząc na wartości obliczonego przyspieszenia przy użyciu metody fourierpower, mogą one być zdumiewające, ale jak już wcześniej wspomnieliśmy, napisana przez nas funkcja przysp działa dobrze tylko na "gładkich" funkcjach. Okazało się również, że **pomiarldrganie9** uzyskuje najmniejszy błąd średniokwadratowy właśnie dla fourierpower. Warto zauważyć też, że dla tego pomiaru fourier oraz fourierpower okazały się tą samą aproksymacją.

Poniżej przedstawiamy wykresy **pomiarladny1**, która dała nam niesatysfakcjonujące wyniki pod względem MSE oraz **pomiardrganie9**, która dała wyjątkowo optymalne wyniki w przypadku fourierpower. Chcieliśmy również porównać wykresy fourier i fourierpower dla tych dwóch pomiarów.

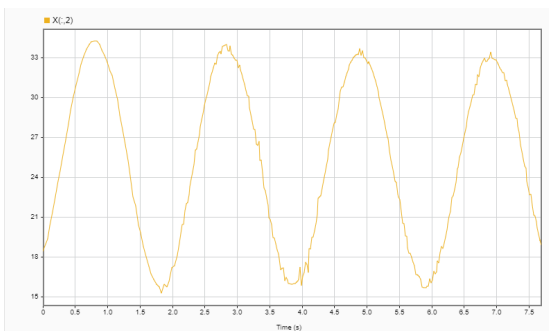
ideat pomiarladny 1



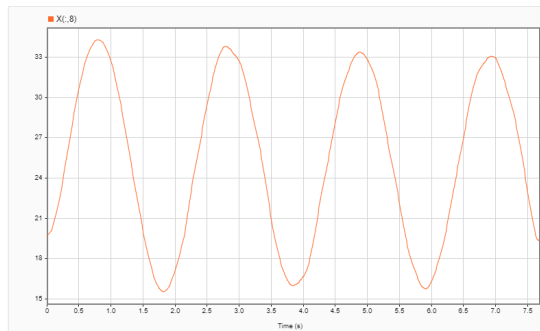
fourierpower pomiarladny1



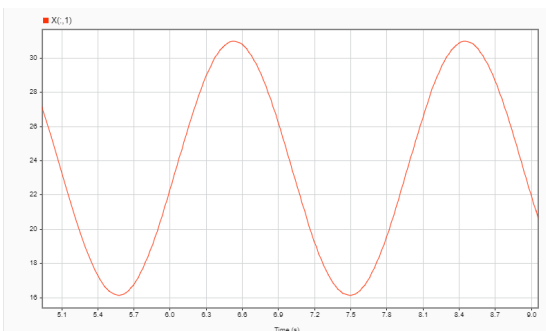
depeak pomiarladny1



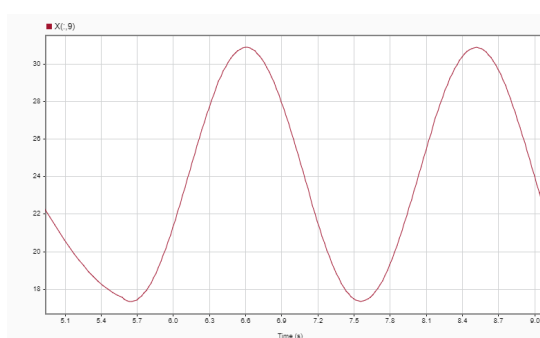
fourier pomiarladny1



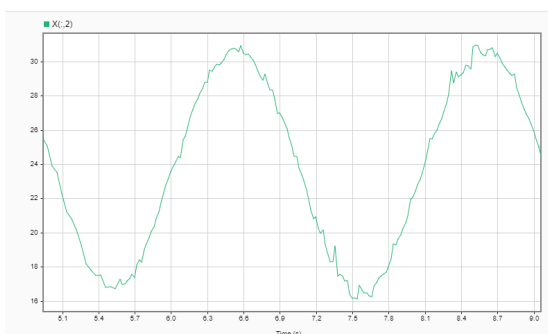
ideat pomiardrganie9



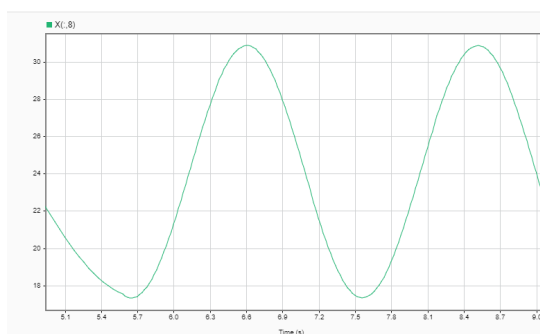
fourierpower pomiardrganie9



depeak pomiardrganie9

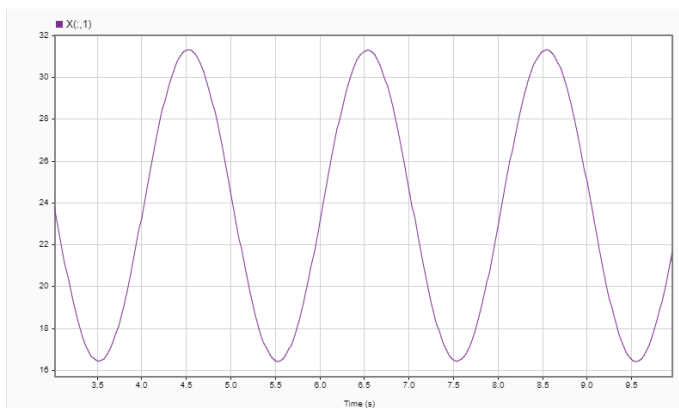


fourier pomiardrganie9

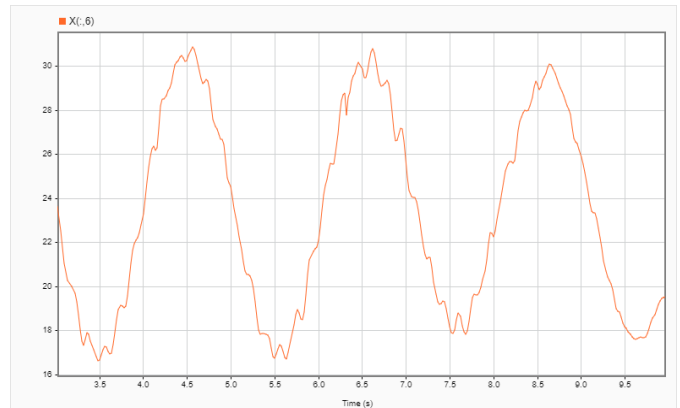


Teraz zwróćmy uwagę na metody o najmniejszej wartości MSE. Możemy zauważyć, że w trzech przypadkach metoda Coiflet wyróżnia się takim współczynnikiem. Ale tak naprawdę, przyglądając się bliżej błędom innych transformat falkowych, możemy dojść do wniosku, że wartości MSE są zbliżone w większości przypadków, więc skorzystanie z którejkolwiek z wybranych metod falkowej aproksymacji dałoby nam podobne efekty. Można jedynie zauważyć, że haar daje nam gorsze wyniki, ale wciąż są one zbliżone do pozostałych.

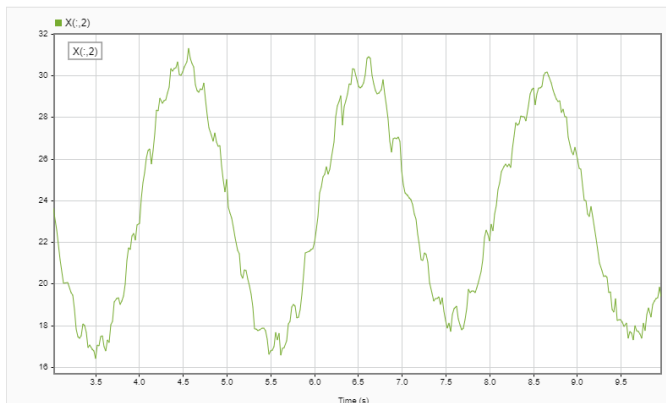
ideat pomiardrgladny3



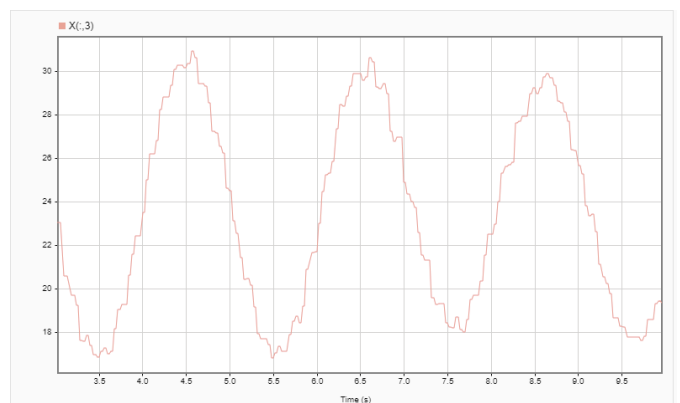
coif pomiardrgladny3



depeak pomiardrgladny3



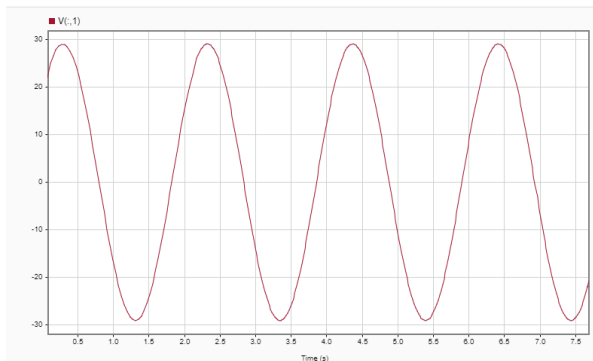
haar pomiardrgladny3



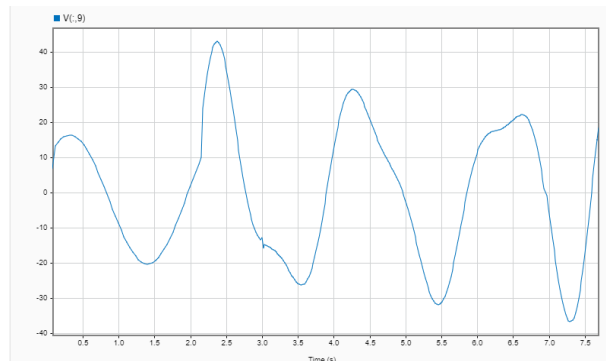
Pod względem entropii warto zaznaczyć, że w przypadku falek Haar miała ona najmniejszą wartość, natomiast największe wartości uzyskaliśmy korzystając z transformaty fourierpower. Z tego wnioskujemy, że tak jak się spodziewaliśmy po analizie MSE oraz wykresu falki Haar i z faktu, że jest ona najprostszą użytą przez nas falką, jej użycie niesie za sobą najmniejszą ilość informacji. W przypadku fourierpower ciężko nam zestawić ze sobą wyniki MSE i entropii.

Teraz przyjrzyjmy się wykresom prędkości oraz ich zmianie w przypadku różnych metod aproksymacji.

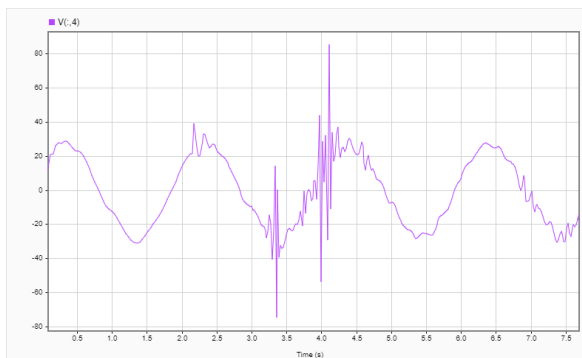
ideał prędkości pomiarładny1



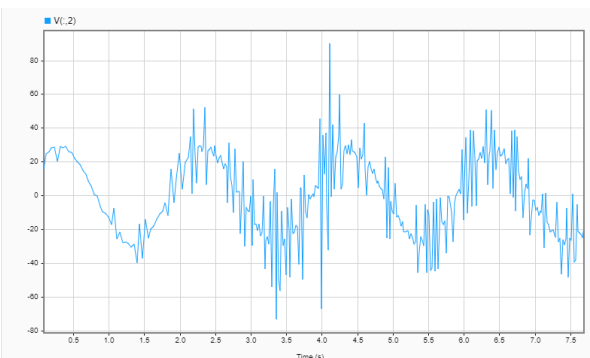
fourierpower prędkości pomiarładny1



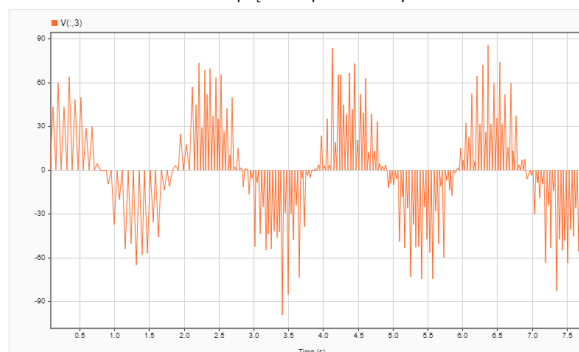
Daubechies prędkość pomiarładny1



depeak prędkość pomiarładny1



Haar prędkość pomiarładny1



Na wykresach powyżej chcieliśmy pokazać, jak zmniejsza się ilość szumów w przypadku metody falkowej z depeak do db oraz z depeak do fourierpower. Chcieliśmy również podkreślić różnice pomiędzy otrzymanymi wykresami a wykresem idealnym wygenerowanym przez nas wzorami. Możemy zauważyć, że wykres fourierpower jest znacznie zniekształcony i nie zachowuje swojego

pierwotnego kształtu. Za to w przypadku metody Daubechies udało się usunąć większość szumów poza tymi największymi i najbardziej widocznymi. Zamieszczamy również wykres falki Haar, który uznaliśmy za ciekawy, ponieważ wykres prędkości Haar jest znacznie mniej czytelny niż wykres prędkości depeak.

## 5. Wnioski

Początkowy wybór różnego rodzaju transformat falkowych okazał się w kontekście naszego projektu nie najlepszym pomysłem. Falki uwydatniają czasowo-częstotliwościowe zmiany sygnały, co pomogłoby zidentyfikować niedoskonałości empirycznego pomiaru, ale nam zależało bardziej na gładkiej charakterystyce naszego ruchu. Falki niejednokrotnie zwiększyły entropię i zmniejszyły popełniany błąd średniokwadratowy, ale nie sprawiły, że zmiany prędkości uwidoczniły się po rzutowaniu zmodyfikowanych danych oraz nie pozwoliły nam skutecznie, maszynowo obliczyć przyspieszenia ziemskiego.

Zgodnie z przypuszczeniami, wybór transformaty Fouriera, której charakterystyka jest zbliżona do istoty problemu, pozwolił przekształcić sygnały w pożądaną przez nas sposób. Ta transformata o bazie w postaci funkcji trygonometrycznych zdecydowanie bardziej przybliżyła charakterystykę badanego ruchu do modelowanego, sinusoidalnego ideału. Transformata fourierowska nie tylko sprawiła, że prędkość przynajmniej na części serii modyfikowanych danych stawała się możliwa do odczytania i przypominała realną, zgodną z teorią zależność, to pozwoliła naszemu autorskiemu algorytmowi odszukać faktyczne maksima lokalne naszych danych, a w efekcie wyznaczyć przyspieszenie ziemskie.

Warto zauważyć, że obydwie sygnały generowane przez transformatę Fouriera różnią się w swojej charakterystyce: **fourier**, który docelowo miał mieć najmniejszy błąd średniokwadratowy, prezentuje nam podobne wartości entropii oraz błędów średniokwadratowego do sygnałów generowanych falkami, jest on również tworzony ze znacznie większej liczby składowych niż **fourierpower**, średnio od 20% do nawet 80% pierwotnych składowych. Z kolei sygnał **fourierpower** charakteryzują zdecydowanie najlepsze osiągi w obliczaniu przyspieszenia ziemskiego oraz relatywnie gładkie wykresy, co jak już wiemy z charakterystyki funkcji obliczającej przyspieszenie ziemskie, jest ze sobą silnie skorelowane. Sygnały te jednak cechują się najwyższą wartością MSE, wszystko to spowodowane jest tym, że z uwagi na wysoki poziom progowania, są one tworzone z bardzo niewielkiej liczby składowych, z reguły poniżej 10, a w wielu przypadkach jedynie 3. Składane funkcje są przyjemne dla oka oraz zapewniają najlepsze pomiary przyspieszenia, lecz są często dalekie od pierwotnego sygnału.

Próbując wyciągnąć wnioski z pomiaru entropii, napotkaliśmy spore trudności. Przy większości pomiarów różnice wartości entropii były nieznaczne, szczególnie wśród falek z różnych rodzin, a także sygnału **fourier**. Ciekawym spostrzeżeniem może być to, że w przypadku sygnału **fourierpower** otrzymaliśmy bardzo niedeterministyczne wyniki, w zależności od pomiaru wartość entropii była z reguły najwyższa bądź najniższa spośród wszystkich sygnałów. W zasadzie jedyną widoczną zależność znaleźliśmy w przypadku odszumiania falką Haara, prawdopodobnie z uwagi na prostą strukturę owej falki oraz nie najlepiej dopasowaną charakterystykę do badanego przez nas problemu dawała ona najniższe spośród wszystkich falek wartości entropii (a także najwyższy błąd średniokwadratowy). Patrząc na wykresy, możemy zauważyć korelację pomiędzy niskimi wartościami entropii, a wyglądem sygnałów generowanych przy użyciu falki Haara - są one widocznie zniekształcone i "kanciaste", szczególnie na tle innych sygnałów.

Analizując wyniki funkcji obliczającej przyspieszenie zauważyliśmy, że dokładność wyniku jest dodatnio skorelowana z częstotliwością próbkowania pomiaru. Wynika to z większej szansy trafienia pomiaru w faktyczną granicę między okresami, a tym samym dokładniejszej wartości obliczonego średniego okresu drgań. Ponadto zauważyliśmy, że wszystkie wyniki funkcji obciążone są błędem (większość oscylowała w okolicy  $10 \frac{m}{s}$ ), co może być spowodowane niedokładnym pomiarem długości wahadła matematycznego (błąd przyspieszenia zależny od błędu pomiaru długości wahadła jest do niego wprost proporcjonalny).

Ciekawą obserwację otrzymaliśmy kreśląc charakterystykę prędkości wahadła po zastosowaniu falki Haara na każdej serii danych. Wykres przybierał niezwykle charakterystyczny kształt, zupełnie niepodobny do innych wykresów danych zmodyfikowanych przy użyciu innych narzędzi, co znacząco zaburzało niesioną przez sygnał wartość informatywną. Nie udało nam się wymyślić żadnego przekonującego wyjaśnienia tego zjawiska. Nasz algorytm oblicza lokalne wartości prędkości na podstawie różnic przebytej pomiędzy punktami drogi  $x$  i czasu  $t$  pomiędzy sąsiednimi pomiarami ( $v = \frac{dx}{dt}$ ). Najprawdopodobniej falka Haara wprowadza zmiany, które niwelują niektóre lokalne różnice wartości, co sprawia, że dane lokalne różnice pomiędzy punktami zanikają, a prędkość zwiększa swoją amplitudę znacząco, często spadając z wartością aż do zera. Falka Haara okazała się, choć interesująca w obserwacji, to bezużyteczna w modelowaniu wartości pochodnych z pierwotnych danych, gdyż zmiany które wprowadza lokalnie zbytnio upraszczają strukturę ruchu.