

Akwizycja danych z sensorów urządzeń mobilnych

Konrad Czaja, Adrian Jałoszewski,
Dorota Kowalik

11 października 2017

1 Cel ćwiczenia

Celem ćwiczenia jest implementacja algorytmu zliczającego kroki na podstawie danych zebranych z czujników urządzenia mobilnego z systemem Android. Algorytm ten powinien być odporny na zachowania inne niż chód oraz możliwy do zastosowania w ciągłej akwizycji danych.

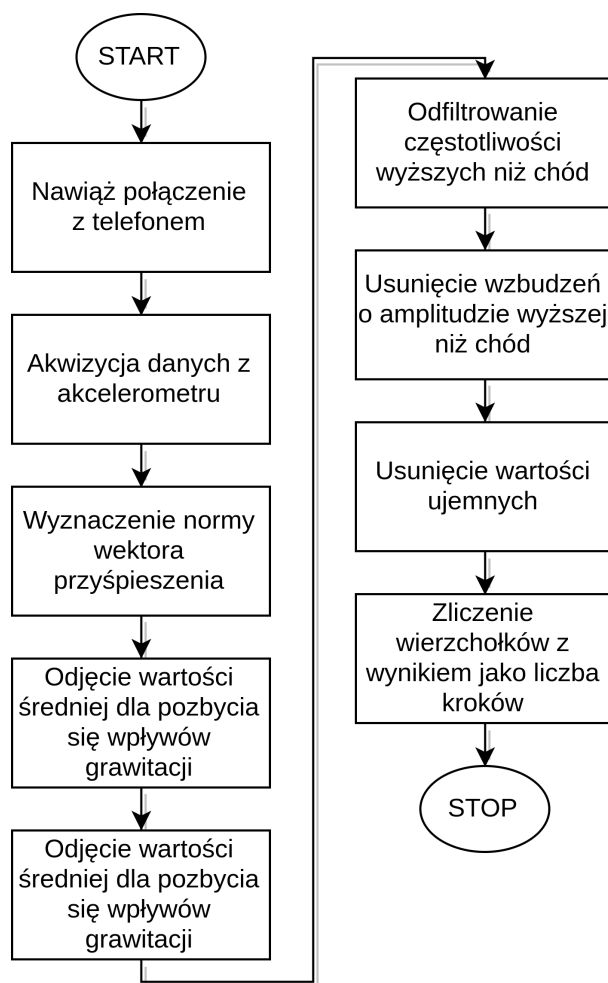
2 Wykonanie ćwiczenia

2.1 Umieszczenie telefonu

Telefon został umieszczony przy szyi tak aby w trakcie obrotów w jak najmniejszym stopniu podlegał sile odśrodkowej w przypadku obrotów.

2.2 Algorytm przetwarzania danych

Przedstawiony algorytm ma zastosowanie dla przypadku kiedy dane z całego przebiegu są już dostępne do analizy.



Rysunek 1: Schemat blokowy algorytmu

2.2.1 Połączenie z telefonem

Połączenie z telefonem zostało nawiązane przez aplikację mobilną pozwalającą na akwizycję danych z urządzenia. Należało ustawić flagę uruchamiającą akcelerometr oraz włączającą logowanie danych. Częstotliwość próbkowania została ustawiona na wysoką – jest to wartość określona w dokumentacji jako 100Hz. Po czasie przeznaczonym na zbieranie danych (w tym przypadku 15s) flaga logowania próbek jest zerowana, a dane są pobierane jako lista składowych przyspieszenia oraz wektor czasu.

```
connector on 'cudo1995';

m = mobiledev
m.AccelerationSensorEnabled = 1;
m.SampleRate = 'high'

disp('Start logging data')
m.Logging = 1;
pause(15)
m.Logging = 0;
disp('Stop logging data')

[a t] = accellog(m);
```

2.2.2 Wyznaczenie wartości przyspieszenia

Dane są umieszczane jako trzy wykresy, z oznaczeniem poszczególnych składowych. Następnie liczona jest wartość przyspieszenia dla każdej próbki.

```
figure(1)
plot(t,a);
legend('X', 'Y', 'Z');
xlabel('Relative time (s)');
ylabel('Acceleration (m/s^2)');

x = a(:,1);
y = a(:,2);
z = a(:,3);
mag = sqrt(sum(x.^2 + y.^2 + z.^2, 2));
```

2.2.3 Wstępna filtracja danych

Pierwszym krokiem w filtracji jest usunięcie składowej stałej. Otrzymany w ten sposób sygnał jest wycentrowany względem zera oraz została od niego odjęta grawitacja. Jako maksymalną częstotliwość chodu zostało przyjęte 3 Hz. Jest to przypadek biegu. Wszystkie częstotliwości powyżej są traktowane jako zakłócenia.

```

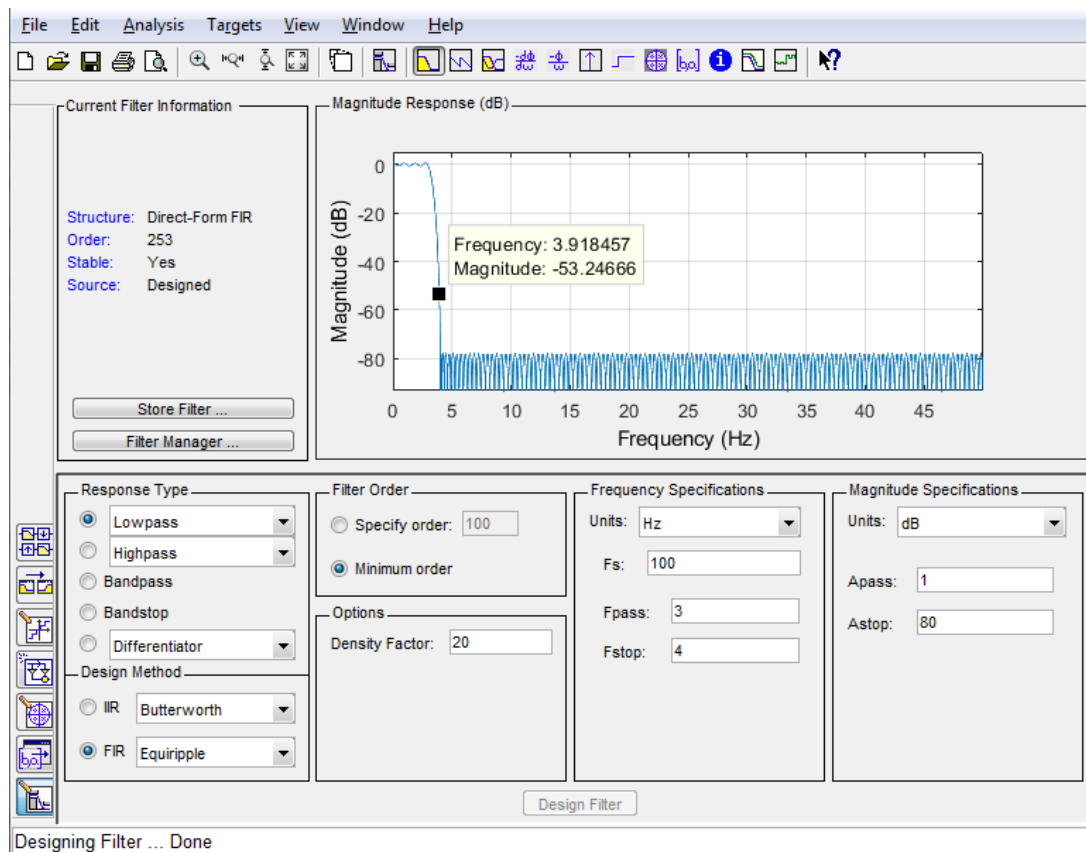
magNoG = mag - mean(mag); % pozbawione grawitacji

a = 1;
b = wsp_filtru;

magNoG = filter(b, a, magNoG);
magFilt = magNoG; % na potrzeby wykresu

```

Współczynnik filtru b został zaimportowany z edytora filtrów:



Rysunek 2: Ustawienia filtru

2.2.4 Usunięcie gwałtownych ruchów

Z zebranych danych wynika, że wszystkie skoki jakie zostały zmierzone mają amplitudę powyżej $6 \frac{m}{s}$ dlatego też takie wartości zostały usunięte z sygnału. Zostało to dokonane przez wyzerowanie wszystkich wartości dodatnich wokół pomiarów przekraczających próg.

```

jump_amplitude_lower_limit = 6;

for i = 1:length(magNoG)
    if (magNoG(i) > jump_amplitude_lower_limit)
        for j = i:length(magNoG)
            if (magNoG(j) > 0)
                magNoG(j) = 0;
            end
        end
    end
end

```

```

        else
            break;
        end
    end
end
for j = (i - 1):-1:1
    if (magNoG(j) > 0)
        magNoG(j) = 0;
    else
        break;
    end
end
end
end
end
end

```

2.2.5 Zliczanie kroków

Zliczanie kroków zostało zrealizowane poprzez zliczenie szczytów w sygnale powstałym przez odfiltrowanie. Wartości ujemne zostały usunięte, gdyż nie są istotne dla dalszych rozważań. Do zliczenia szczytów została wykorzystana funkcja `findpeaks` z odchyleniem standardowym sygnału wynikowego jako minimalna wysokość szczytu.

```

magNoG = max(magNoG, 0);
magFilt = max(magFilt, 0);

minPeakHeight = std(magNoG);
[pks, locs] = findpeaks(magNoG, 'MINPEAKHEIGHT', ...
    minPeakHeight);
numSteps = numel(pks)

```

2.2.6 Wyrysowanie wyników

```

figure(2)
plot(t, mag)
xlabel('Time (s)');
ylabel('Acceleration (m/s^2)');

figure()
plot(t, magFilt)
xlabel('Time (s)');
ylabel('Acceleration (m/s^2) with steps and jumps - filtered');

figure()
plot(t, magNoG);
xlabel('Time (s)');
ylabel('Acceleration (m/s^2)');

hold on

```

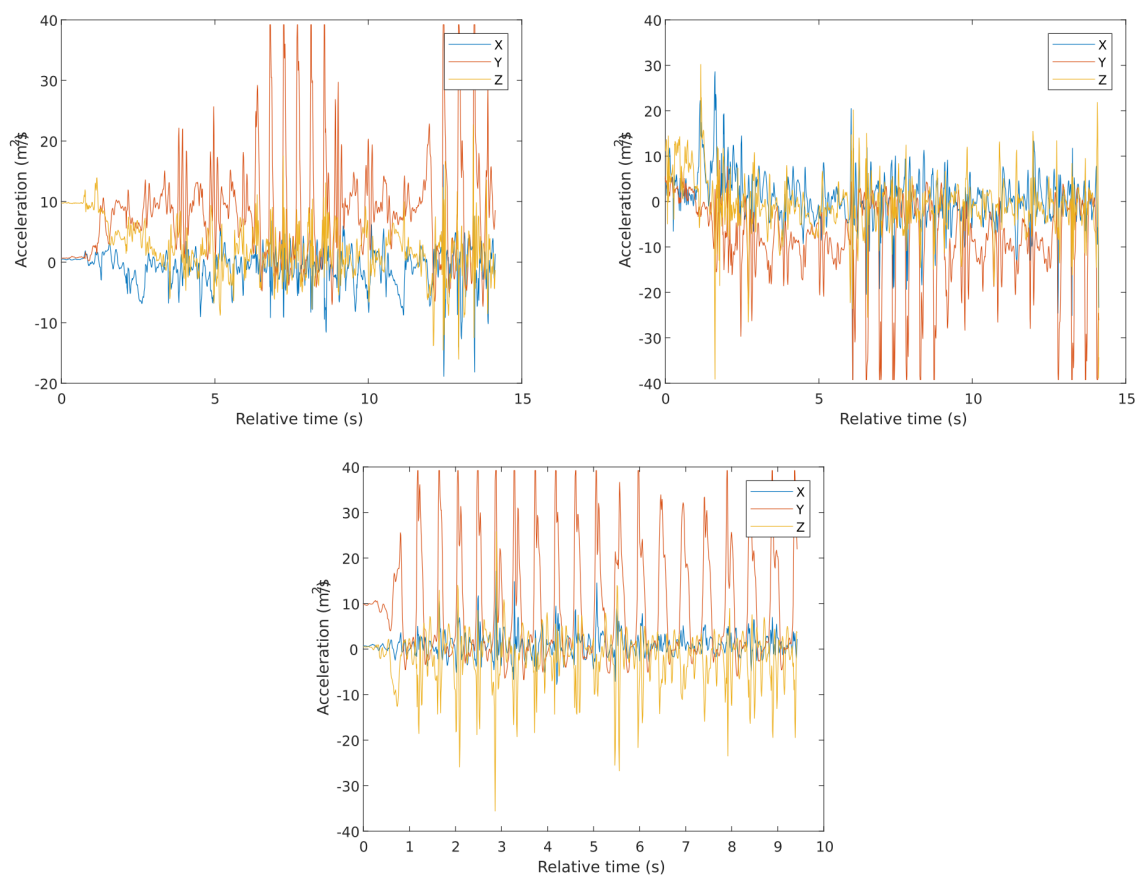
```

plot(t(locs), pks, 'r', 'Marker', 'v', 'LineStyle', 'none');
title('Counting Steps and Jumping');
xlabel('time (s)');
ylabel('Acceleration magnitude, no gravity (m/s^2)');
hold off;

```

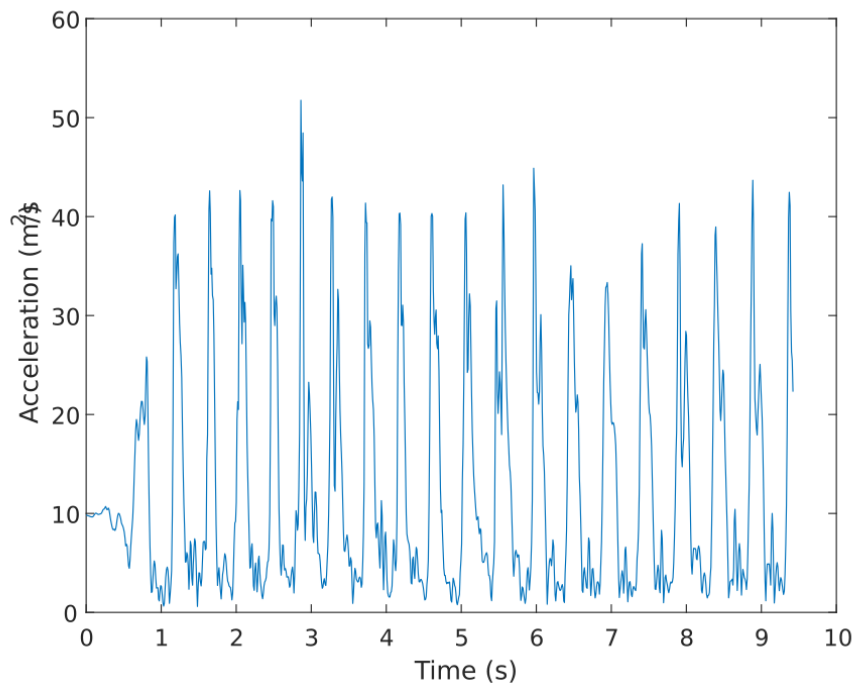
3 Pomiary oraz ich analiza

Dane zebrane z czujników przed przetworzeniem są przebiegami składowych przyspieszenia zarejestrowanego przez akcelerometry. Zależnie od położenia przebiegi te posiadają składową stałą.



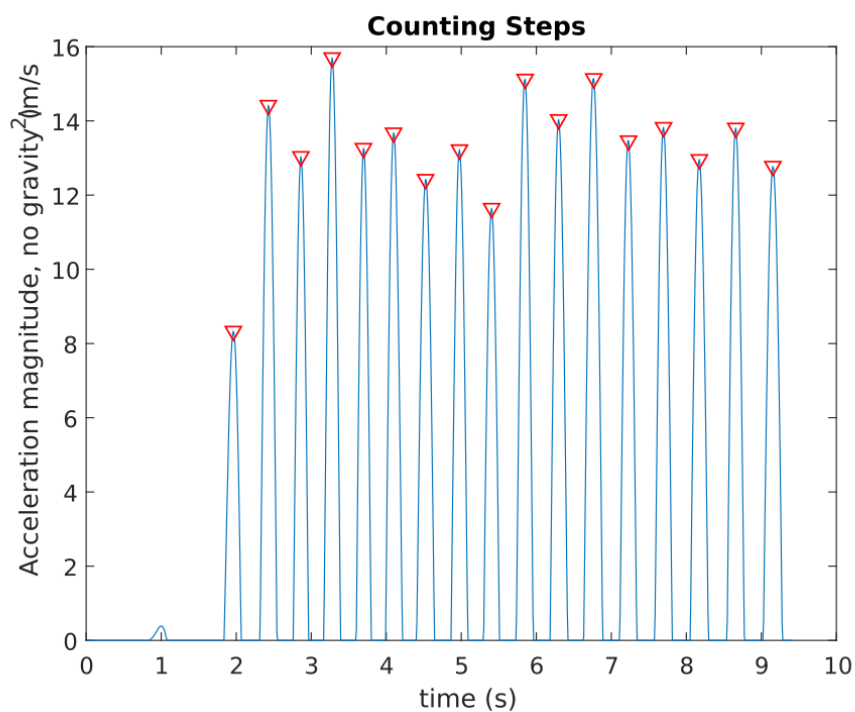
Rysunek 3: Dane zebrane z czujników

Dla przypadku zawierającego tylko podskoki wartość przyspieszenia kształtowała się następująco:



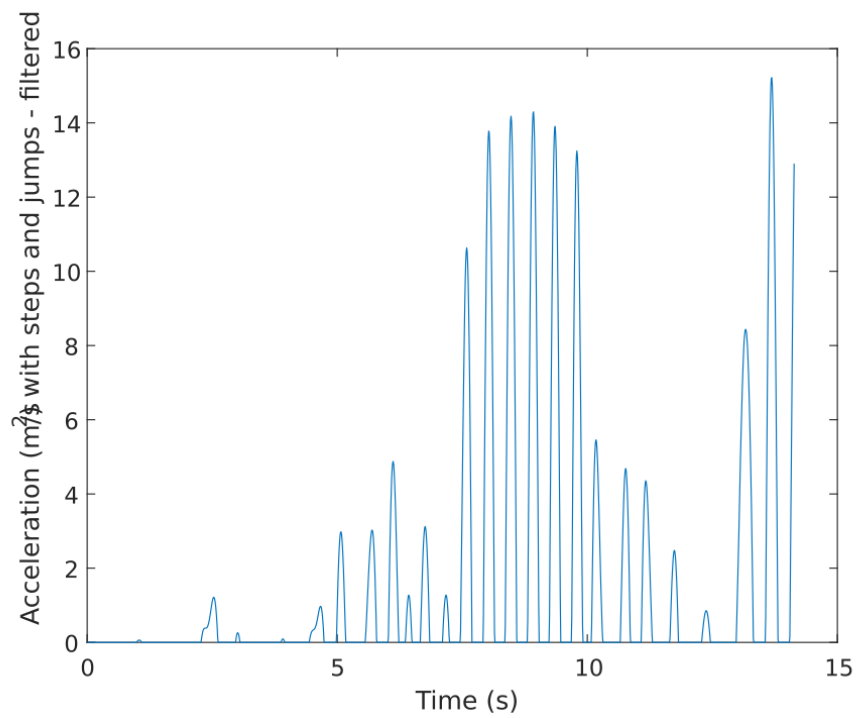
Rysunek 4: Wartość przyspieszenia przy samych podskokach

Po zastosowaniu algorytmu dla przebiegu z pominięciem kroku usuwającego podskoki sygnał kształtuje się następująco:



Rysunek 5: Zliczone i zaznaczone rozpoznane podskoki

W przypadku, gdzie występowały zarówno podskoki jak i zwykły chód przebieg po zastosowaniu algorytmu bez usuwania podskoków kształtuje się następująco:



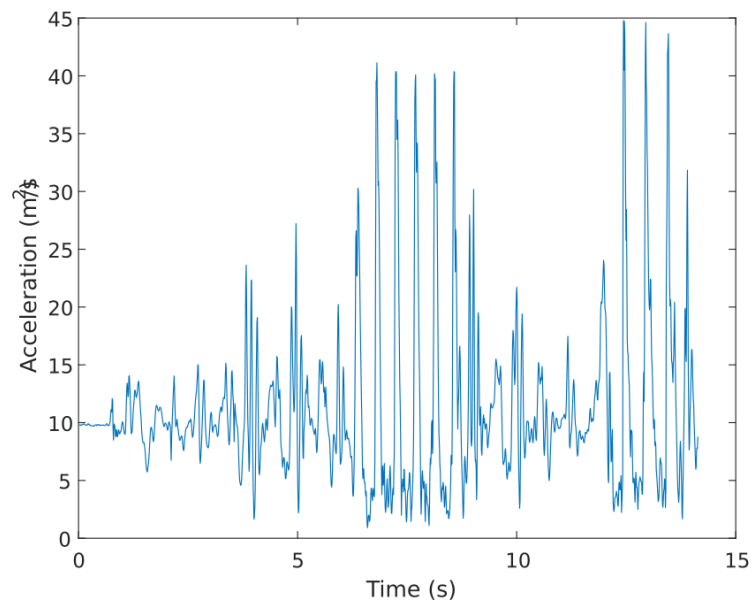
Rysunek 6: Kroki i podskoki

Na podstawie pomiarów nasunął się wniosek przyjęcia rozgraniczenia kroków i podskoków progiem $6 \frac{\text{m}}{\text{s}}$. Podobne rezultaty były uzyskiwane dla innych przemarszy z podskokami.

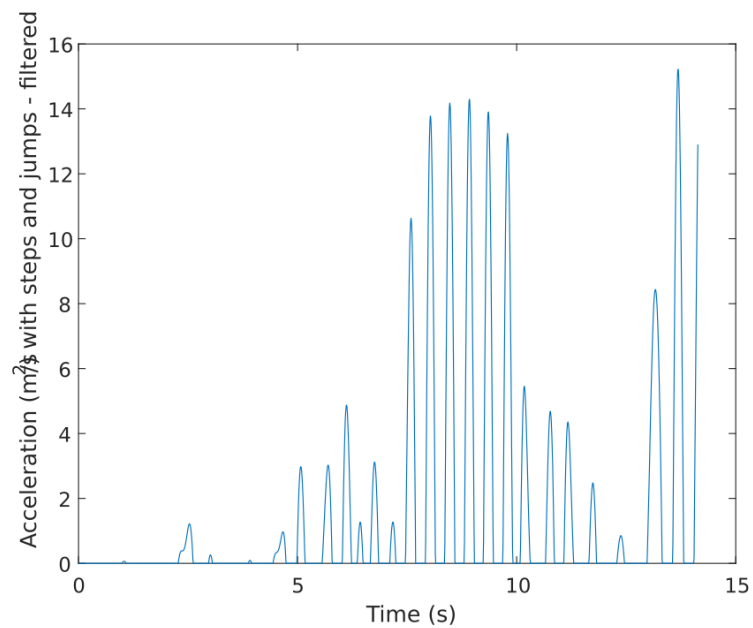
4 Wyniki

Rozwiązanie zostało przetestowane dla dwóch przemarszy z podskokami różniącymi się tym czy najpierw następuje marsz czy podskoki.

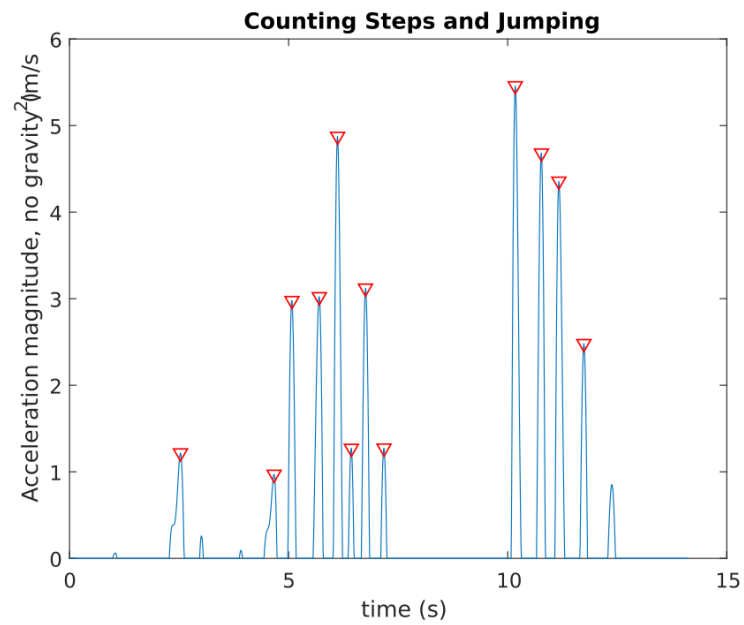
4.1 marsz – podskoki – marsz – podskoki



Rysunek 7: Przed filtrowaniem



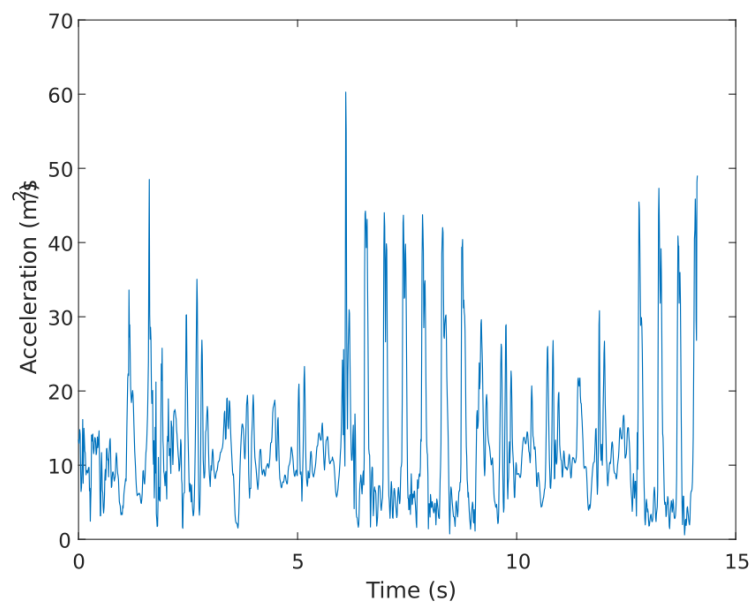
Rysunek 8: Bez wycięcia podskoków



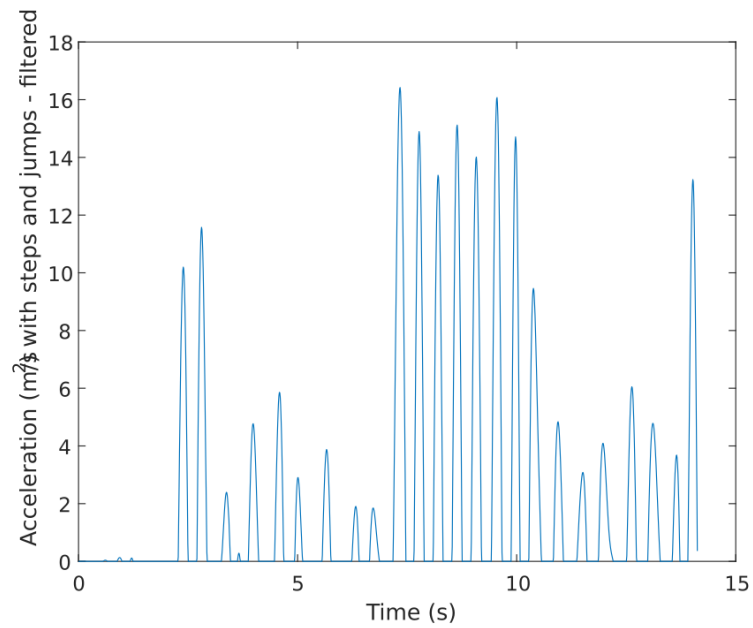
Rysunek 9: Po wycięciu podskoków

Zostało wykrytych 12 kroków.

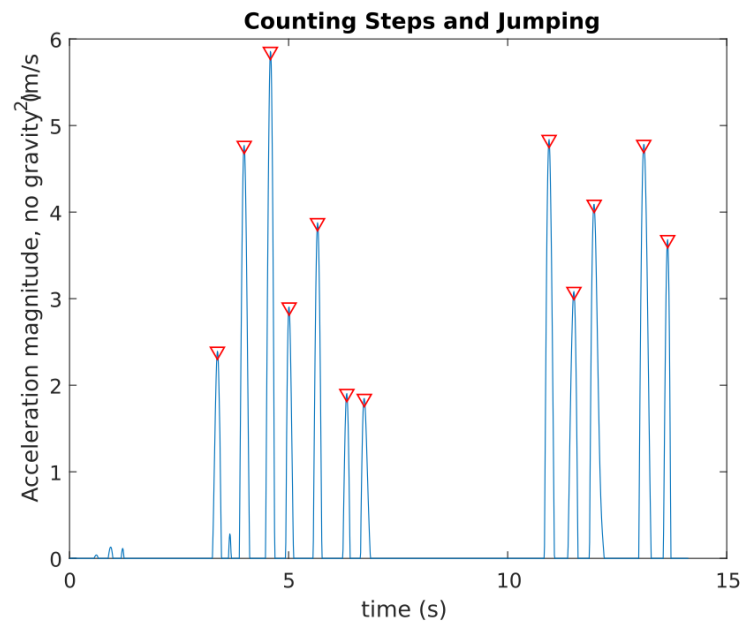
4.2 podskoki – marsz – podskoki – marsz



Rysunek 10: Przed filtrowaniem



Rysunek 11: Bez wycięcia podskoków



Rysunek 12: Po wycięciu podskoków

Zostało wykrytych 12 kroków.

5 Podsumowanie i obserwacje

Ze względu na naturę zastosowanego przez nas filtru początkowe pomiary zostały usunięte. Dlatego też był potrzebny pewien czas rozbiegu dla programu po którym dopiero zaczęły napływać wyniki możliwe do dalszej obróbki.

Po rozbiegu program poprawnie liczy kroki, skutecznie pomijając podskoki. Zostało to zweryfikowane poprzez zliczanie kroków i porównanie uzyskanej liczby z uzyskanym w programie wynikiem.

Zaproponowany algorytm nie bierze pod uwagę przypadku gdy mamy do czynienia z nietypowym, skocznym chodem. W tym przypadku część kroków może być sklasyfikowanych jako podskoki i są usuwane.