



**Politechnika  
Śląska**

## **PRACA MAGISTERSKA**

Estymacja kroków z czujników ruchu

**Maksymilian KISIEL**  
**Nr albumu: 296829**

**Kierunek:** Informatyka  
**Specjalność:** Interaktywna Grafika Trójwymiarowa

### **PROWADZĄCY PRACĘ**

**Dr hab. inż. Agnieszka Szczęsna, prof. PŚ.**

**KATEDRA Grafiki, Wizji Komputerowej i Systemów Cyfrowych**  
**Wydział Automatyki, Elektroniki i Informatyki**

**Gliwice 2025**



## **Tytuł pracy**

Estymacja kroków z czujników ruchu

## **Streszczenie**

Celem pracy jest analiza i porównanie algorytmów detekcji kroków na podstawie pomiarów czujników inercyjnych. Zaprojektowano stanowisko pomiarowe oparte na mikrokontrolerze Raspberry Pi Pico z dwoma czujnikami IMU. Zaimplementowano pięć algorytmów obejmujących metody progowe, analizę spektralną, systemy adaptacyjne oraz fuzję danych sensorycznych. Wykonano pomiary w różnych scenariuszach ruchu i punktach montażu sensora.

Uzyskane wyniki pozwalają na ocenę skuteczności algorytmów w zależności od typu aktywności i lokalizacji sensora. Montaż na ramieniu okazał się najskuteczniejszy, algorytmy spektralne najlepsze dla schodów. Test TUG stanowił największe wyzwanie.

## **Słowa kluczowe**

estymacja kroków, czujniki inercyjne, przetwarzanie sygnałów, systemy wbudowane, porównanie algorytmów

## **Thesis title**

Step estimation from motion sensors

## **Abstract**

The aim of this study is to analyze and compare step detection algorithms using measurements from inertial sensors. A measurement system based on Raspberry Pi Pico microcontroller with two IMU sensors was designed. Five algorithms were implemented, including threshold-based methods, spectral analysis, adaptive systems, and sensor data fusion. Measurements were conducted across various motion scenarios and sensor mounting points.

The obtained results allow for assessment of algorithm effectiveness depending on activity type and sensor location. Arm mounting proved most effective, spectral algorithms best for stairs. The TUG test posed the greatest challenge.

## **Key words**

step detection, inertial measurement unit, signal processing, embedded systems, algorithm comparison



# Spis treści

<b>1 Wstęp</b>	<b>1</b>
1.1 Cel pracy . . . . .	2
1.2 Zakres pracy . . . . .	3
1.3 Zwięzła charakterystyka rozdziałów . . . . .	4
1.4 Wkład autora . . . . .	5
<b>2 Analiza tematu</b>	<b>7</b>
2.1 Czujniki inercyjne IMU . . . . .	8
2.1.1 Budowa i zasada działania . . . . .	8
2.1.2 Parametry techniczne wpływające na detekcję kroków . . . . .	10
2.1.3 Problemy techniczne i ich wpływ na detekcję . . . . .	10
2.2 Algorytmy klasyczne detekcji kroków . . . . .	11
2.2.1 Metody progowe (ang. <i>Threshold-based</i> ) . . . . .	11
2.2.2 Metoda oparta na wykrywaniu przejścia przez zero (ang. <i>Zero-Crossing</i> ) . . . . .	13
2.2.3 Analiza częstotliwościowa . . . . .	14
2.3 Metody zaawansowane . . . . .	15
2.3.1 Aktualizacja zerowej prędkości (ang. <i>Zero Velocity Update</i> ) . . . . .	15
2.3.2 Progi adaptacyjne . . . . .	16
2.3.3 Dopasowywanie wzorca (ang. <i>Template matching</i> ) . . . . .	17
2.4 Głębokie uczenie w detekcji kroków . . . . .	18
2.4.1 Konwolucyjne sieci neuronowe . . . . .	18
2.4.2 Rekurencyjne sieci neuronowe . . . . .	19
2.4.3 Transformery . . . . .	20
2.4.4 Uczenie transferowe (ang. <i>Transfer Learning</i> ) i uczenie maszynowe na urządzeniach brzegowych . . . . .	21
2.5 Wpływ lokalizacji na dokładność detekcji . . . . .	22
2.5.1 Hierarchia dokładności według lokalizacji czujnika . . . . .	22
2.5.2 Wpływ prędkości ruchu na dokładność . . . . .	23
2.5.3 Implikacje praktyczne i rekomendacje . . . . .	23
2.6 Wyzwania i ograniczenia obecnych rozwiązań . . . . .	24

2.6.1	Problemy dobrze rozwiązane . . . . .	24
2.6.2	Otwarte wyzwania . . . . .	24
2.6.3	Luki metodologiczne . . . . .	26
2.7	Zastosowania praktyczne . . . . .	26
2.7.1	Inteligentne zegarki i opaski sportowe . . . . .	26
2.7.2	Nawigacja wewnętrzna . . . . .	27
2.7.3	Monitoring medyczny i rehabilitacja . . . . .	28
2.8	Kierunki rozwoju i przyszłe badania . . . . .	28
2.9	Podsumowanie . . . . .	29
<b>3</b>	<b>Przedmiot pracy</b>	<b>31</b>
3.1	Koncepcja rozwiązania . . . . .	31
3.1.1	Architektura systemu . . . . .	31
3.1.2	Przyjęte założenia projektowe . . . . .	32
3.2	Wybór i uzasadnienie metod algorytmicznych . . . . .	33
3.2.1	Detekcja szczytów (ang. <i>Peak Detection</i> ) . . . . .	33
3.2.2	Przejście przez zero (ang. <i>Zero-Crossing</i> ) . . . . .	33
3.2.3	Analiza spektralna (ang. <i>Spectral Analysis</i> ) . . . . .	34
3.2.4	Próg adaptacyjny (ang. <i>Adaptive Threshold</i> ) . . . . .	34
3.2.5	Estymator odchylenia kierunku kroku (ang. <i>Step Heading Offset Estimator</i> ) . . . . .	35
3.2.6	Uzasadnienie różnorodności metod . . . . .	35
3.3	Wybór platformy sprzętowej . . . . .	36
3.3.1	Raspberry Pi Pico 2W - specyfikacja i możliwości . . . . .	36
3.3.2	Uzasadnienie wyboru . . . . .	36
3.3.3	Alternatywy i ich ograniczenia . . . . .	37
3.4	Czujniki inercyjne IMU . . . . .	37
3.4.1	Moduł 1: <i>Waveshare Pico-10DOF-IMU</i> . . . . .	37
3.4.2	Moduł 2: <i>Adafruit ST 9-DoF</i> . . . . .	38
3.4.3	Porównanie modułów i uzasadnienie wykorzystania dwóch platform	40
3.4.4	Konfiguracja sensorów w systemie . . . . .	40
3.5	Oprogramowanie i narzędzia . . . . .	41
3.5.1	Oprogramowanie mikrokontrolera - MicroPython . . . . .	41
3.5.2	Aplikacja analityczna - Python z bibliotekami naukowymi . . . . .	42
3.5.3	System kontroli wersji - Git i GitHub . . . . .	42
3.6	Metodologia zbierania i analizy danych . . . . .	43
3.6.1	Proces pomiarowy . . . . .	43
3.6.2	Scenariusze testowe . . . . .	44
3.6.3	Metryki oceny jakości . . . . .	45

3.6.4	Interpreter danych zewnętrznych . . . . .	46
3.7	Analiza teoretyczna rozwiązania . . . . .	47
3.7.1	Przygotowanie sygnału . . . . .	47
3.7.2	Założenia upraszczające . . . . .	47
3.7.3	Złożoność obliczeniowa . . . . .	48
3.7.4	Wrażliwość na parametry . . . . .	48
3.8	Wkład pracy i jej oryginalność . . . . .	48
3.8.1	Unikalne cechy systemu . . . . .	49
3.8.2	Potencjalne kierunki rozbudowy . . . . .	49
3.9	Podsumowanie . . . . .	50
<b>4</b>	<b>Badania</b>	<b>51</b>
4.1	Stanowisko badawcze . . . . .	51
4.1.1	Dokumentacja fotograficzna stanowiska . . . . .	52
4.1.2	Scenariusze testowe . . . . .	54
4.2	Metodyka badań . . . . .	56
4.2.1	Protokół eksperymentalny . . . . .	56
4.2.2	Proces oznaczania kroków . . . . .	57
4.2.3	Dobór parametrów algorytmów . . . . .	59
4.2.4	Implementacja algorytmów . . . . .	59
4.3	Struktura danych eksperymentalnych . . . . .	62
4.3.1	Format nagrań pomiarowych . . . . .	62
4.3.2	Przetwarzanie wstępne . . . . .	63
4.3.3	Metryki oceny jakości detekcji . . . . .	63
4.4	Wyniki eksperymentów . . . . .	65
4.4.1	Analiza przykładowego nagrania . . . . .	65
4.4.2	Porównanie wyników algorytmów . . . . .	65
4.4.3	Porównanie globalne algorytmów . . . . .	67
4.4.4	Wyniki szczegółowe dla wszystkich kombinacji . . . . .	68
4.4.5	Zestawienie zbiorcze wyników . . . . .	69
4.4.6	Wpływ punktu montażu na skuteczność detekcji . . . . .	70
4.5	Analiza wyników . . . . .	71
4.5.1	Skuteczność poszczególnych algorytmów . . . . .	71
4.5.2	Wpływ scenariusza ruchu . . . . .	72
4.5.3	Porównanie sensorów . . . . .	73
4.5.4	Ograniczenia przeprowadzonych badań . . . . .	73
4.6	Wnioski . . . . .	74

<b>5 Podsumowanie</b>	<b>77</b>
5.1 Zrealizowane zadania . . . . .	77
5.2 Wnioski końcowe . . . . .	78
5.3 Możliwości rozwoju i kierunki dalszych badań . . . . .	79
5.4 Realizacja celów pracy . . . . .	80
<b>Bibliografia</b>	<b>85</b>
<b>Spis skrótów i symboli</b>	<b>89</b>
<b>Lista dodatkowych plików, uzupełniających tekst pracy</b>	<b>93</b>
<b>Spis rysункów</b>	<b>95</b>
<b>Spis tabel</b>	<b>97</b>

# Rozdział 1

## Wstęp

Analiza ruchu, w tym ruchu człowieka, stanowi jedno z najstarszych zagadnień badawczych w historii nauki. Wszystkie dobrowolne ruchy, w tym chód, są wynikiem skomplikowanego procesu, w którym biorą udział mózg, rdzeń kręgowy, nerwy obwodowe, mięśnie, kości i stawy. Aby dokładnie zrozumieć ten proces, sposób jego analizy i co może pójść nie tak, wymagane jest podstawowe zrozumienie trzech naukowych dziedzin: anatomicznej, fizjologii i biomechaniki [49]. Już w IV wieku przed naszą erą, Arystoteles zaczął opisywać ciała zwierząt jako systemy mechaniczne. Badał on również ruch mięśni i poddawał je analizie geometrycznej. Na przestrzeni kolejnych stuleci osiągnięto znaczący postęp zarówno w rozumieniu anatomicznej i fizjologicznej ciała ludzkiego - czego przykładem są pionierskie prace Leonardo da Vinci - jak i w metodach analizy ruchu człowieka. Od studiów Eadwearda Muybridge'a<sup>1</sup> z 1878 roku, przez "licznik 10 000 kroków"<sup>2</sup> (jap. *manpo-kei*) z lat 60. XX wieku, aż po współczesne systemy inercyjne - ludzkość nieustannie doskonali systemy pomiarowe, a co za tym idzie, metody zliczania kroków. Współczesne mikro-elektromechaniczne systemy (ang. *micro-electromechanical systems, MEMS*) rewolucjonizowały tę dziedzinę, umożliwiając pomiary z dokładnością rzędu "*mili-g*" przy częstotliwościach pomiarowych sięgających kilkuset herców [1, 22].

Paradoksalnie, im bardziej zaawansowane technologicznie stają się nasze urządzenia pomiarowe, tym trudniejsze okazuje się osiągnięcie uniwersalnej precyzji w detekcji tak podstawowego zjawiska jak ludzki krok. Czujniki inercyjne (ang. *Inertial Measurement Unit, IMU*) są w stanie generować tysiące próbek na sekundę, ale przetwarzanie takiej ilości danych na prosty licznik kroków pozostaje wyzwaniem.

Estymacja kroków z czujników ruchu znajduje się na przecięciu kilku dziedzin nauki, m. in. przetwarzania sygnałów cyfrowych, biomechaniki i inżynierii systemów wbudowanych. Ta interdyscyplinarność sprawia, że optymalizacja algorytmów detekcji kroków wymaga nie tylko zrozumienia podstaw filtracji i analizy spektralnej, ale także intuicji do-

<sup>1</sup>Brytyjski fotograf, który zasłynął innowacyjnymi badaniami ruchu za pomocą sprzężonych aparatów fotograficznych. Fotografie przedstawiały kolejne fazy ruchu galopującego konia [40].

<sup>2</sup>Mechaniczne pedometry produkowane przez japońską firmę *Yamasa* [29].

tyczącej fizjologii chodu ludzkiego oraz ograniczeń obliczeniowych platform wbudowanych (ang. *embedded systems*, np. *Raspberry Pi Pico 2 W*).

Jak wskazują liczne prace naukowe, rozwój metod detekcji kroków opiera się na balansowaniu pomiędzy prostotą obliczeniową a odpornością na zmienność ludzkiego chodu. Proste podejścia, takie jak metody progowe (wykrywanie przekroczenia ustalonego progu sygnału), czy metody oparte na dopasowaniu do wzorca kroku (ang. *template-based matching*), są energooszczędne i łatwe do implementacji, lecz szybko tracą skuteczność w warunkach odbiegających od laboratoryjnych [34, 50]. Z kolei metody oparte na uczeniu maszynowym i sieciach neuronowych oferują większą odporność na różnorodność chodu i ułożenie sensora, jednak kosztem większej złożoności obliczeniowej i zapotrzebowania na odpowiednie dane treningowe [45].

Pomimo tak szerokiego spektrum podejść, literatura nadal dostarcza fragmentarycznych wyników, uzyskiwanych w odmiennych warunkach eksperymentalnych, z różnymi czujnikami, częstotliwościami próbkowania i metodologiami walidacji. Utrudnia to obiektywną ocenę skuteczności poszczególnych algorytmów, a wpływ charakterystyk sprzętowych sensorów IMU na jakość detekcji wciąż pozostaje relatywnie mało zbadanym obszarem, mimo jego kluczowego znaczenia dla projektowania systemów wbudowanych.

## 1.1 Cel pracy

Głównym celem pracy jest kompleksowa analiza i obiektywne porównanie różnych algorytmów estymacji kroków na podstawie danych z czujników inercyjnych.

W ramach tego celu praca skoncentruje się na budowie systemu pomiarowego, który pozwoli na zebranie surowych danych, oraz przygotowaniu programu umożliwiającego analizę i porównanie osiąganych wyników. Implementacja aplikacji analizującej ma obejmować także moduł interpretera, który będzie w stanie przetłumaczyć dane z odrębnych badań do formatu wykorzystywanego przez program. Dzięki temu możliwe będzie porównanie wyników na podstawie danych z różnych źródeł - pomiary ze znaną liczbą kroków (znany punkt odniesienia, ang. *ground truth*) i nieznaną. Cele szczegółowe obejmują:

1. Zaprojektowanie i implementacja stanowiska pomiarowego opartego na platformie Raspberry Pi Pico 2W, demonstrującego możliwości współczesnych mikrokontrolerów w aplikacjach analizy ruchu.
2. Implementacja i optymalizacja pięciu fundamentalnie różnych podejść do detekcji kroków: algorytmów progowych (ang. *Peak Detection*), analizy przejść przez zero (ang. *Zero Crossing*), metod częstotliwościowych (ang. *Spectral Analysis*), systemów adaptacyjnych (ang. *Adaptive Threshold*) oraz technik łączenia danych z wielu sensorów (ang. *SHOE*). Wybór obejmuje reprezentację głównych nurtów metodologicznych w dziedzinie.

3. Przeprowadzenie analizy porównawczej w kontrolowanych scenariuszach testowych, obejmujących podstawowe wzorce chodu człowieka: chód w tempie naturalnym, przyspieszonym i trucht oraz specyficzne przypadki - poruszanie się po schodach w górę i w dół w tempie naturalnym. Analiza ma na celu identyfikację mocnych i słabych stron poszczególnych algorytmów w kontekście różnorodności wzorców kinematycznych.

## 1.2 Zakres pracy

Zadaniem pracy jest porównanie jakości poszczególnych algorytmów do detekcji kroków na podstawie pomiarów z czujników inercyjnych w różnych scenariuszach testowych. Zakres pracy obejmuje następujące obszary:

1. Przygotowanie i implementacja systemu pomiarowego

Układ powinien pozwalać na zebranie surowych danych z czujników inercyjnych, a także na przesłanie ich do programu analizującego.

2. Przygotowanie programu umożliwiającego analizę

Program powinien umożliwić analizę i porównanie jakości detekcji, wraz z wizualnym przedstawieniem wskaźników jakości poszczególnych algorytmów.

3. Przeprowadzenie pomiarów

Przeprowadzenie pomiarów w różnych scenariuszach testowych i różnych punktów mocowania urządzenia pomiarowego.

4. Interpreter danych z odrębnych badań

Przetłumaczenie danych z odrębnych badań (bez punktu odniesienia) do formatu wykorzystywanego przez program analizujący w celu porównania wyników.

5. Porównanie algorytmów detekcji

Porównanie jakości poszczególnych algorytmów detekcji kroków na podstawie danych z czujników inercyjnych.

6. Analiza wpływu charakterystyk sprzętowych i parametrów detekcji

Analiza wpływu charakterystyk sprzętowych czujników IMU i parametrów poszczególnych algorytmów na jakość detekcji kroków.

## 1.3 Zwięzła charakterystyka rozdziałów

Praca składa się z pięciu rozdziałów merytorycznych oraz dodatków.

**Rozdział 1 - Wstęp** przedstawia wprowadzenie w problem detekcji kroków z czujników inercyjnych, osadzenie zagadnienia w kontekście historycznym i naukowym oraz formułuje cel i zakres pracy. Rozdział określa również wkład autora w realizację poszczególnych elementów systemu.

**Rozdział 2 - Przegląd literatury** zawiera kompleksowy przegląd istniejących metod detekcji kroków, od prostych algorytmów progowych, przez metody oparte na przetwarzaniu sygnałów (analiza częstotliwościowa, przejścia przez zero), aż po zaawansowane podejścia wykorzystujące uczenie maszynowe i sieci neuronowe. Rozdział omawia również metryki oceny jakości algorytmów oraz charakterystyki czujników IMU istotne dla detekcji kroków.

**Rozdział 3 - Przedmiot i metody badań** opisuje zaprojektowany system pomiarowy oparty na Raspberry Pi Pico 2W oraz dwa czujniki IMU (Waveshare MPU9250 i Adafruit LSM6DSOX+LIS3MDL). Przedstawiono architekturę oprogramowania, protokoły komunikacji bezprzewodowej oraz szczegółową specyfikację pięciu zaimplementowanych algorytmów detekcji. Rozdział definiuje również metodologię eksperymentów i proces walidacji wyników.

**Rozdział 4 - Wyniki badań** prezentuje szczegółowe rezultaty 84 sesji pomiarowych przeprowadzonych w siedmiu scenariuszach testowych i czterech punktach montażu sensora. Przedstawiono porównanie skuteczności algorytmów za pomocą metryk F1-score, precyzji i czułości, analizę wpływu punktu montażu i scenariusza ruchu oraz dyskusję mocnych i słabych stron poszczególnych metod detekcji.

**Rozdział 5 - Podsumowanie** podsumowuje wykonane prace, formułuje wnioski końcowe oraz wskazuje kierunki dalszego rozwoju. Rozdział potwierdza realizację wszystkich założonych celów pracy i identyfikuje praktyczne zastosowania wyników badań.

Dodatki zawierają spis skrótów i symboli wykorzystanych w pracy oraz listę dodatkowych plików (kod źródłowy, zbiory danych, dokumentacja).

## 1.4 Wkład autora

Autor pracy odpowiedzialny był za takie elementy jak:

1. Projektowanie architektury systemu pomiarowego

Autor odpowiadał za projektowanie ogólnej architektury systemu pomiarowego. Obejmowało to określenie struktury oprogramowania, interfejsów komunikacyjnych i organizacji komponentów programu.

2. Implementacja i optymalizacja interfejsu komunikacji

Autor zaimplementował system umożliwiający komunikację pomiędzy urządzeniem pomiarowym i programem do analizy w oparciu o moduł Wi-Fi i protokoły UDP i HTTP, a także optymalizację tego procesu w celu minimalizacji opóźnień i maksymalizacji częstotliwości próbkowania.

3. Przygotowanie programu

Autor zdefiniował i zaimplementował pięć algorytmów estymacji, które wykorzystywane są przez program umożliwiający analizę i porównanie jakości detekcji, wraz z wizualnym przedstawieniem wskaźników jakości poszczególnych algorytmów. Program zawiera moduł interpretera, który tłumaczy dane z odrębnych badań do formatu wykorzystywanego w pracy.

4. Przeprowadzenie pomiarów

Autor przeprowadzał pomiary w różnych scenariuszach testowych i różnych punktów mocowania urządzenia pomiarowego.

5. Dokumentacja i wnioski

Wkład autora w tym obszarze obejmuje analizę wyników, wnioski płynące z przeprowadzonych badań oraz propozycje kierunków dalszego rozwoju systemu. Autor był również odpowiedzialny za stworzenie dokumentacji technicznej, która pozwala na reprodukcję i uruchamianie środowiska.



## Rozdział 2

### Analiza tematu

Detekcja kroków na bazie pomiarów z czujników inercyjnych stanowi fundamentalny problem w dziedzinie analizy aktywności człowieka (ang. *Human Activity Recognition, HAR*) i znajduje szerokie zastosowanie w nawigacji wewnętrznej, monitorowaniu zdrowia oraz różnych aplikacjach sportowych.

Problem ten sprowadza się do identyfikacji kluczowych zdarzeń w cyklu chodu, który składa się z dwóch głównych faz: fazy podporu (ang. *stance phase*), podczas której stopa ma kontakt z podłożem, oraz fazy przenoszenia (ang. *swing phase*), kiedy stopa jest w powietrzu [18]. Najistotniejszymi zdarzeniami w tym cyklu są *heel strike* (HS) - moment uderzenia piętą o podłożę, oraz *toe-off* (TO) - moment oderwania palców od podłożu. Precyzyjna detekcja tych zdarzeń umożliwia nie tylko liczenie kroków, ale także estymację parametrów czasowo-przestrzennych chodu, takich jak długość kroku, kadencja czy asymetria.

Te możliwości analizy parametrów chodu znajdują praktyczne zastosowanie w szerskim zakresie dziedzin. W opiece zdrowotnej algorytmy detekcji wykorzystywane są do monitorowania pacjentów z chorobą Parkinsona, gdzie długość kroku jest silnie powiązana z zaawansowaniem choroby [36], oraz w rehabilitacji po osteoartrozie stawu kolanowego [17, 34]. W kontekście nawigacji wewnętrznej (ang. *Pedestrian Dead Reckoning, PDR*), gdzie sygnał GPS jest niedostępny, precyzyjne liczenie kroków w połączeniu z estymacją długości kroku i kierunku ruchu pozwala na wyznaczenie trajektorii użytkownika [26]. Urządzenia do monitorowania aktywności fizycznej (ang. *fitness tracking*), takie jak smartwatche czy opaski sportowe, bazują na algorytmach detekcji kroków do szacowania dziennej aktywności fizycznej właściciela.

Kluczowym wyzwaniem w detekcji kroków jest osiągnięcie wysokiej dokładności we wszystkich warunkach życia codziennego. Prędkość chodu znaczaco wpływa na charakterystykę sygnału - bardzo wolny chód (poniżej  $2 \frac{km}{h}$ ) generuje sygnały, które są trudne do odróżnienia od szumu z powodu małych amplitud, podczas gdy bieg (powyżej  $9 \frac{km}{h}$ ) charakteryzuje się dużymi amplitudami i krótkimi fazami podporu [35]. Lokalizacja czujnika na ciele wprowadza kolejny problem do tego zagadnienia - różnice w dokładności detekcji

między montażem na kostce, biodrach, albo nadgarstku mogą sięgać od 10 do 30 punktów procentowych [15, 35, 42]. Warunki środowiskowe, takie jak nieregularny teren, schody, czy zakręty, dodatkowo komplikują kwestie detekcji.

Do oceny skuteczności algorytmów detekcji kroków stosuje się metryki obejmujące tradycyjne miary klasyfikacji binarnej: precyza (ang. *precision*), czułość (ang. *recall*), F1-score (średnią harmoniczną precyzji i czułości), oraz dokładność (ang. *accuracy*). W kontekście liczenia kroków często wykorzystuje się MAPE (ang. *Mean Absolute Percentage Error*), czyli średni bezwzględny błąd procentowy względem rzeczywistej liczby kroków. W zastosowaniach wymagających detekcji konkretnych zdarzeń chodu (HS, TO), istotny jest również błąd czasowy mierzony w milisekundach lub procentach cyklu chodu [51].

Przeglądając dostępne w literaturze badania, możemy zauważać wyraźną ewolucję podejść do problemu detekcji kroków - od prostych metod progowych lat 2000., przez zaawansowane metody wykorzystujące transformacje falkowe (ang. *wavelet*) i ZUPT (ang. *Zero Velocity Update*) w latach 2010., aż po współczesne rozwiązania oparte na głębokim uczeniu (ang. *Deep Learning*). Pomimo znaczącego postępu, współczesne systemy wciąż zmagają się z wyzwaniemiami związanymi z detekcją w warunkach rzeczywistych, szczególnie dla chodu nieregularnego, wolnych prędkości oraz urządzeń noszonych na nadgarstku.

## 2.1 Czujniki inercyjne IMU

Czujniki inercyjne (ang. *Inertial Measurement Unit, IMU*) stanowią podstawowy element wykorzystywany w detekcji kroków i ogólnym badaniu aktywności człowieka. Typy czujnik IMU składa się z trzech głównych komponentów mierzących różne aspekty ruchu w przestrzeni trójwymiarowej:

- akcelerometr,
- żyroskop,
- magnetometr.

Każdy z tych komponentów dostarcza odmiennych informacji o charakterze ruchu, co łącznie umożliwia pełną analizę dynamiki chodu.

### 2.1.1 Budowa i zasada działania

Dla akcelerometru, żyroskopu i magnetometru definiujemy wektory pomiarowe:

$$\mathbf{a} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}, \quad \boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \quad \mathbf{m} = \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix}, \quad (2.1)$$

gdzie:

- a** - wektor przyspieszenia liniowego (akcelerometr),
- $\omega$**  - wektor prędkości kątowej (żyroskop),
- m** - wektor pola magnetycznego (magnetometr),
- $x, y, z$  - składowe w układzie kartezjańskim.

Sygnał z czujnika w czasie przedstawiamy jako sekwencję dyskretnych pomiarów:

$$\mathbf{s} = \{ \mathbf{s}[0], \mathbf{s}[1], \mathbf{s}[2], \dots, \mathbf{s}[N-1] \}, \quad (2.2)$$

gdzie  $\mathbf{s}[i]$  oznacza wartość sygnału w  $i$ -tej próbce czasowej ( $i = 0, 1, \dots, N-1$ ),  $N$  to całkowita liczba próbek, a próbki są pobierane z częstotliwością  $f_s$ , Hz.

**Akcelerometr** mierzy przyspieszenie liniowe wzdłuż trzech osi prostopadłych. W kontekście detekcji kroków, pionowa składowa przyspieszenia (oś z,  $a_z$ ) zawiera szczególnie istotną informację o zdarzeniach *heel strike i toe-off* (HS, TO), które prezentowane są jako charakterystyczne szczyty i doliny w sygnale [18]. Typowe zakresy pomiarowe akcelerometrów używanych w aplikacjach związanych z chodem wynoszą do  $\pm 16 g$  ( $\approx \pm 157 \frac{m}{s^2}$ ). Podczas normalnego chodu amplitudy przyspieszenia rzadko przekraczają  $\pm 4 g$ , jednak podczas biegania czy skoków mogą osiągać wartości do  $\pm 8 g$ .

Warto zauważyć, że akcelerometr rejestruje nie tylko przyspieszenia pochodzące od ruchu, lecz także składową wynikającą z działania grawitacji. Obecność tej składowej jest z jednej strony użyteczna, ponieważ pozwala na orientacyjną kalibrację i określenie położenia czujnika względem wektora pionu, lecz z drugiej strony stanowi problem podczas analizy dynamicznych zmian ruchu. W praktyce wymaga to odjęcia przyspieszenia ziemskiego, co bywa źródłem błędów, zwłaszcza gdy orientacja czujnika nie jest stała.

**Żyroskop** mierzy prędkość kątową wokół trzech osi. W analizie chodu żyroskop dostarcza do algorytmów estymujących rotację w stawach i orientację segmentów ciała. Szczególnie istotna jest prędkość kątowa wokół osi poprzecznej ciała (ang. *medial-lateral axis*<sup>1</sup>), która wykazuje charakterystyczne wzorce podczas fazy przenoszenia odnóża [23]. Typowe zakresy pomiarowe wynoszą od  $\pm 250 \frac{\text{deg}}{\text{s}}$  do  $\pm 2000 \frac{\text{deg}}{\text{s}}$ , przy czym dla chodu człowieka wystarczające są zakresy do  $\pm 500 \frac{\text{deg}}{\text{s}}$ .

**Magnetometr** mierzy natężenie pola magnetycznego Ziemi, co umożliwia wyznaczenie orientacji urządzenia względem kierunków geograficznych. W kontekście detekcji kroków magnetometr odgrywa rolę pomocniczą i jest wykorzystywany głównie w systemach nawigacji wewnętrznej do wyznaczania kierunku ruchu [28]. Pomiary magnetometru są szczególnie podatne na zakłócenia od urządzeń elektronicznych i konstrukcji stalowych, co ogranicza ich użyteczność.

<sup>1</sup>Oś poprzeczna (medialno-lateralna) - przebiega od jednej strony ciała do drugiej (np. od prawej do lewej strony). Wyznacza kierunek przyśrodkowy (łac. *directio medialis*) oraz kierunek boczny (łac. *directio lateralis*) [37]

W praktycznych zastosowaniach do detekcji kroków wykorzystuje się czujniki IMU o częstotliwości próbkowania (ang. *sampling rate*)  $100\text{ Hz}$ , zakresie akcelerometru  $\pm 160 \frac{\text{m}}{\text{s}^2}$  i żyroskopu  $\pm 2000 \frac{\text{deg}}{\text{s}}$ . Czujniki takie, jak XSens MTw Awinda, Bosch BMI270 czy ActiGraph GT3X+, często posiadają wbudowane algorytmy liczenia kroków i rozpoznawania kontekstu aktywności, co czyni je popularnymi w badaniach klinicznych [2, 12, 38].

### 2.1.2 Parametry techniczne wpływające na detekcję kroków

**Częstotliwość próbkowania** (ang. *sampling rate*) jest krytycznym parametrem wpływającym na dokładność detekcji kroków. Badania wykazują, że częstotliwość  $\geq 20\text{ Hz}$  jest absolutnym minimum - poniżej tej wartości wydajność algorytmów drastycznie spada, często poniżej 70 % skuteczności [4]. Dla podstawowego rozpoznawania chodu wystarczające jest próbkowanie  $10\text{ Hz}$ , które jednocześnie minimalizuje zużycie baterii urządzenia, jednak aplikacje wymagające szczegółowej analizy wykorzystują częstotliwości z zakresu  $50\text{-}100\text{ Hz}$ , które stanowią standard w badaniach klinicznych [34]. Monitorowanie wydajności sportowej może wymagać jeszcze wyższych częstotliwości, sięgających  $208\text{ Hz}$  [19].

Zgodnie z tw. **Nyquista-Shannona**, częstotliwość próbkowania powinna być co najmniej dwukrotnie wyższa od maksymalnej częstotliwości sygnału. Typowa częstotliwość kroku podczas chodu mieści się w zakresie od  $0.5$  do  $2\text{ Hz}$  (od  $\approx 30$  do  $\approx 120 \frac{\text{kroków}}{\text{min}}$ ), co sugerowałoby, że próbkowanie na poziomie  $4\text{ Hz}$  jest wystarczające. Jednakże harmoniczne wyższych rzędów oraz konieczność precyzyjnego wykrywania zdarzeń *heel strike* i *toe-off* uzasadniają stosowanie znacznie wyższych częstotliwości, aby zminimalizować ryzyko pomylenia właściwego sygnału z szumem.

Oprócz częstotliwości próbkowania, istotny wpływ na dokładność detekcji ma również jakość samych czujników. Dokładność akcelerometrów jest bardzo zróżnicowana - od najtańszych czujników mikro-elektromechanicznych (ang. *micro-electromechanical systems, MEMS*) z błędem  $100\text{ mg}$  do precyzyjnych jednostek z błędem rzędu  $10\text{ }\mu\text{g}$ . Ten kompromis między kosztem, rozmiarem, masą i dokładnością jest szczególnie istotny w kontekście urządzeń mobilnych, gdzie wysokiej klasy IMU byłyby niepraktyczne ze względu na wymogi energetyczne i gabarytowe [28].

### 2.1.3 Problemy techniczne i ich wpływ na detekcję

**Dryft żyroskopu** stanowi fundamentalne ograniczenie w kontekście długoterminowej dokładności systemów inercyjnych. Dryft jest powolnym przesuwaniem się wartości zeroowej żyroskopu, powodujące, że całkowanie prędkości kątowej w celu uzyskania orientacji prowadzi do błędu rosnącego liniowo z czasem. Jest on dominującym czynnikiem wpływającym na dokładność systemów nawigacji wspieranej metodą ZUPT (metoda aktualizacji zerowej prędkości, sekcja 2.3.1). W praktyce oznacza to, że bez zewnętrznej referencji (takiej jak momenty zerowej prędkości w metodzie ZUPT), estymacja orientacji może być

błędna o dziesiątki stopni po kilku minutach ruchu [8]. Problem ten jest znacznie mniej istotny, a czasami wręcz pomijalny, dla badań składających się z krótkich eksperymentów (poniżej  $\approx 2$  minut).

**Szum akcelerometru** wynika z wpływu środowiska, sił zewnętrznych oraz niestabilności komponentów elektronicznych. Jest on szczególnie problematyczny w kontekście słabych sygnałów charakterystycznych dla bardzo wolnego chodu, gdzie amplitudy przyspieszenia związane z chodem są porównywalne z poziomem szumu. Strategia filtrowania zazwyczaj obejmuje filtr pasmowo-przepustowy w zakresie 0.25-2.5 Hz, który pozwala na skuteczną separację rzeczywistego sygnału od szumu [11, 35].

**Efekty temperaturowe** wprowadzają dodatkową zmienność. Bias i czułość zarówno akcelerometrów jak i żyroskopów zmieniają się wraz z temperaturą, co wymaga odpowiedniej kompensacji lub częstej kalibracji. Niektóre czujniki, np. Bosch BMI330 zostały zaprojektowane specjalnie dla aplikacji wymagających pracy w podwyższonych temperaturach do  $105^{\circ}\text{C}$ , co jest istotne w kontekście urządzeń noszonych na ciele, które mogą szybko się nagrzewać poprzez bezpośredni kontakt ze skórą, np. w trakcie intensywnego wysiłku.

Powszechnie występujące **zakłócenia magnetyczne** znacząco ograniczają przydatność magnetometrów. Obecność urządzeń elektronicznych, konstrukcji stalowych oraz pól magnetycznych z instalacji elektrycznych powoduje, że pomiary magnetometru mogą być istotnie zniekształcone, wymagając dopasowania współczynnika interferencji [28].

Opisane problemy techniczne mogą być częściowo kompensowane poprzez odpowiednie metody kalibracji. Standardowe podejścia obejmują kalibrację fabryczną oraz kalibrację wykonywaną przez użytkownika, takie jak korekcja błędu dla trzyosiowego akcelerometru czy dopasowanie elipsoidy dla magnetometru. Skuteczne są również metody łączące dane z wielu czujników (ang. *sensor fusion*) [33]. Należy jednak podkreślić, że niniejsza praca koncentruje się na metodach detekcji, które nie wymagają kalibracji przed użyciem, co ma kluczowe znaczenie dla praktycznych zastosowań konsumenckich.

## 2.2 Algorytmy klasyczne detekcji kroków

### 2.2.1 Metody progowe (ang. *Threshold-based*)

Metody progowe stanowią fundament większości komercyjnych rozwiązań do detekcji kroków ze względu na niską złożoność obliczeniową, co umożliwia działanie w czasie rzeczywistym. W tej klasie algorytmów kluczowym wyzwaniem jest dobór odpowiedniego progu, który musi być wystarczająco czuły, aby wykryć faktyczne kroki, jednocześnie eliminując fałszywe detekcje spowodowane szumem lub ruchami niezwiązanymi z chodem np. uderzenie w urządzenie pomiarowe lub wibracje spowodowane tym, że urządzenie nie przylega ciasno do ciała.

Podstawowe podejście polega na analizie wielkości przyspieszenia (wyznaczenie długości wektora przyspieszenia)

$$a_{mag}[i] = \sqrt{a_x[i]^2 + a_y[i]^2 + a_z[i]^2} \quad (2.3)$$

oraz detekcji momentów, w których sygnał przekracza zadany próg. Ducharme i współpracownicy przedstawili metodę wykorzystującą filtrowanie pasmowo-przepustowe (ang. *bandpass filter*) w zakresie  $0.25\text{-}2.5\text{ Hz}$  oraz technikę wybierania szczytów (ang. *peak picking*) [11]. Badanie na 75 osobach dorosłych z czujnikami montowanymi na nadgarstku i w talii wykazało wysoką dokładność dla różnych aktywności. Kluczowe znaczenie ma odejmowanie średniej wartości przyspieszenia w celu kompensacji wpływu orientacji urządzenia na działanie algorytmu - bez tego kroku błędy detekcji znacząco rosły.

Znaczący postęp stanowi zastosowanie **progów dynamicznych** (ang. *Adaptive Threshold*) zamiast statycznych. W dokumentacji technicznej dla czujnika ADXL367 zaprezentowanej przez *Analog Devices* przedstawiono implementację osiągającą średnią dokładność na poziomie  $\approx 97\%$  przy częstotliwości próbkowania  $50\text{ Hz}$  [2]. Algorytm analizuje sumę wartości bezwzględnych przyspieszenia z trzech osi:

$$a[i] = |a_x[i]| + |a_y[i]| + |a_z[i]|, \quad (2.4)$$

co zapewnia odporność na zmienną orientację czujnika. Próg dynamiczny  $T[i]$  jest adaptowany w oparciu o średnią pozycję między lokalnymi maksimami i minimami wykrytych kroków:

$$T[i] = \frac{1}{N} \sum_{i=1}^N \left( \frac{\max_i + \min_i}{2} \right), \quad (2.5)$$

gdzie  $N = 4$  jest rozmiarem bufora progów, a  $\max_i$  i  $\min_i$  to wartości szczytowe i minimalne  $i$ -tego wykrytego kroku. Dodatkowo algorytm weryfikuje, czy amplituda kroku przekracza stały próg czułości ( $0.1\text{ g}$  dla zakresu  $\pm 2\text{ g}$ ). Dynamiczny próg działa lepiej niż stały próg szczególnie przy zmiennych warunkach użytkowania - w przeprowadzonych eksperymentach statyczny próg generował znacząco więcej fałszywych detekcji podczas przejść między różnymi aktywnościami.

Skończona maszyna stanów (ang. *Finite State Machine, FSM*) jako wariant metod progowych wprowadza koncepcję stanów dyskretnych reprezentujących różne fazy chodu [4]. Typowa implementacja obejmuje następujące stany: *standing*, *step\_detected*, *swing\_phase*, *stance\_phase*. Przejścia między stanami są definiowane przez progi oraz warunki czasowe, co redukuje fałszywe detekcje poprzez wymaganie spełnienia sekwencji warunków zamiast pojedynczego przekroczenia progu.

Niezależnie od konkretnego wariantu, praktyczne implementacje metod progowych muszą uwzględniać szereg dodatkowych mechanizmów: filtrowanie wstępne dla eliminacji

szumu (zazwyczaj filtr pasmowo przepustowy w zakresie 0.25-2.5 Hz), histerezę progową dla uniknięcia wielokrotnych detekcji tego samego kroku, minimalny odstęp czasowy między kolejnymi krokami (zazwyczaj 0.3-0.5 s) oraz okno czasowe dla adaptacji dynamicznego progu (zazwyczaj 3-10 s).

### 2.2.2 Metoda oparta na wykrywaniu przejścia przez zero (ang. *Zero-Crossing*)

Algorytmy oparte na wykrywaniu przejść przez zero (ang. *zero-crossing*) umożliwiają nie tylko liczenie kroków, ale również precyzyjną detekcję konkretnych zdarzeń w cyklu chodu. Metoda ta wykorzystuje fakt, że określone składowe przyspieszenia lub prędkości kątowej przechodzą przez wartość zerową w charakterystycznych momentach cyklu chodu.

Zastosowanie przejścia przez zero w przypadku pacjentów z protezami kończyn dolnych wykazało przewagę nad wykrywaniem pików (ang. *peak detection*). Metoda okazała się skuteczniejsza od wykrywania pików w 65 % przypadków amputacji podudzia i 81 % amputacji uda. Wykorzystano pojedynczy czujnik IMU w okolicy lędźwiowo-biodrowej, analizując przyspieszenie wzdłużne (ang. *fore-aft acceleration*). Wyniki są szczególnie istotne, ponieważ pacjenci z protezami wykazują asymetryczny wzorzec chodu stanowiący wyzwanie dla standardowych algorytmów [6].

Zastosowanie ciągłej transformaty falkowej (ang. *Continuous Wavelet Transform, CWT*, sekcja 2.2.3) do detekcji kontaktu początkowego (ang. *Initial Contact, IC*) i kontaktu końcowego (ang. *Final Contact, FC*) z pionowego przyspieszenia stanowi zaawansowaną metodę implementacji algorytmu przejścia przez zero [18]. Metodologia obejmuje całkowanie transformacji falkowej przyspieszenia pionowego, a następnie różniczkowanie przy użyciu falki gaussowskiej. *IC* identyfikowane jest jako lokalne minimum, podczas gdy *FC* jako lokalne maksimum w przetworzonym sygnale. Walidacja względem systemu laboratoryjnego (np. GaitRite<sup>2</sup>) wykazała doskonałą zgodność dla parametrów przestrzenno-czasowych chodu. Krytyczne ograniczenie stanowią błędy na poziomie milisekund znacząco wpływające na zgodność pomiarów zmienności i asymetrii - chociaż średnie wartości są dokładne, analiza stabilności kroku wymaga wyższej precyzji czasowej.

Wykorzystanie wyłącznie żyroskopu zamiast akcelerometru w połączeniu z przejściem przez zero i metodą progową wykazało wyższą czułość przy wolnym chodzie (poniżej  $2 \frac{km}{h}$ ) [23]. Podejście to oferuje adaptacyjność do różnych orientacji urządzenia, co jest szczególnie wartościowe w aplikacjach wykorzystujących smartfony, gdzie orientacja urządzenia jest nieprzewidywalna i może się znacząco zmieniać w czasie.

Wybór składowej sygnału do analizy przejścia przez zero ma krytyczne znaczenie. Przyspieszenie pionowe (ang. *vertical*) jest optymalne dla detekcji *heel strike* przy mon-

---

<sup>2</sup>Popularny i przenośny system służący do analizy chodu za pomocą chodnika badającego siłę nacisku <https://www.gaitrite.com/>

tażu na stopie, podczas gdy przyspieszenie wzdłużne (ang. *fore-aft*) jest preferowane dla czujników na tułowiu. Prędkość kątowa wzduż osi poprzecznej ciała (ang. *medial-lateral*) wykazuje szczególną czułość przy wolnym chodzie. Prędkość liniowa, uzyskana po całkowaniu przyspieszenia, jest wykorzystywana szczegółowo w metodach ZUPT.

### 2.2.3 Analiza częstotliwościowa

Metody spektralne oferują odporność na zakłócenia i niezależność od miejsca montażu czujnika wykorzystując fakt, że chód człowieka charakteryzuje się periodycznością reprezentowaną przez wyraźne piki w dziedzinie częstotliwości.

Szybka transformata Fouriera (ang. *Fast Fourier Transform, FFT*) na prędkościach kątowych z żyroskopu, z identyfikacją chodu przez piki spektralne w okolicy  $\approx 1\text{ Hz}$  (normalna częstotliwość chodu) osiągnęło  $\approx 93\%$  średniej dokładności [27]. Podejście przewyższało metody oparte na akcelerometrze o co najmniej  $\approx 3\%$  oraz produkty komercyjne (Pacer, Spring Rain) o co najmniej  $\approx 5\%$ . Istotną zaletą jest działanie niezależnie od umiejscowienia urządzenia - algorytm zachowuje dokładność niezależnie od tego, czy telefon znajduje się w ręce, kieszeni, czy różnych innych pozycjach.

Alternatywne wykorzystanie *FFT* do wygładzania danych akcelerometru i wykrywania sygnałów periodycznych osiągnęło 87.52 % dokładności [10]. Metoda ta dobrze radzi sobie ze zmianami trybu aktywności (chód, stanie, siedzenie), automatycznie adaptując się do różnych wzorców ruchu poprzez analizę dominujących częstotliwości w oknie czasowym, o ile są to wzorce ściśle periodyczne. Ograniczenie stanowi chodzenie po schodach z przerwami, gdzie periodyczność jest zakłócona.

Krótkookresowa transformata Fouriera (ang. *Short-Time Fourier Transform, STFT*) stanowi rozszerzenie *FFT* umożliwiające analizę czasowo-częstotliwościową. Algorytm dzieli sygnał na krótkie segmenty i stosuje *FFT* do każdego oddzielnego, pozwalając na śledzenie zmian częstotliwości w czasie. W kontekście porównania algorytmów *STFT* osiągnęło imponujące 99.57 % TPR<sup>3</sup> z 0 % FPR<sup>4</sup> dla czujnika zamocowanego na stopie [4].

Ciągła transformata falkowa z falkami Gaussa dla detekcji zdarzeń *HS* i *TO* z przyspieszenia wzdłużnego wykazała wysoką dokładność w walidacji względem systemu optoelektronicznego Vicon<sup>5</sup>. Badanie na 11 pacjentach z chorobą Parkinsona i 12 osobach starszych ujawniło, że różne typy falki są optymalne dla różnych warunków chodu (*gaus2*<sup>6</sup> dla skrętów, *db2*<sup>7</sup> dla chodu prostoliniowego) [36].

Optymalizacja skali falkowej wymaga uwzględnienia prędkości chodu. Badania wyka-

<sup>3</sup>ang. *True Positive Rate* - odsetek prawdziwych pozytywów, czułość

<sup>4</sup>ang. *False Positive Rate* - odsetek fałszywych pozytywów

<sup>5</sup>Vicon T10 - System do analizy ruchu (motion capture) wykorzystujący 10 kamer, który służy do precyzyjnego śledzenia i analizy ruchu obiektów i ludzi w czasie rzeczywistym

<https://www.vicon.com/applications/life-sciences/clinical-gait-analysis/>

<sup>6</sup>Falka Gaussa drugiego stopnia

<sup>7</sup>Falka Daubechies drugiego stopnia

zały silną korelację między prędkością chodu a częstotliwością kroków ( $R^2 = 0.92$ ), co umożliwia automatyczne dostrojenie parametrów algorytmu. Dla detekcji kontaktu końcowego (FC) optymalna skala silnie zależy od częstotliwości kroków ( $p \ll 0.001$ ), podczas gdy detekcja kontaktu początkowego (IC) jest mniej wrażliwa na ten parametr [7].

Praktyczna implementacja metod częstotliwościowych wymaga uwzględnienia szeregu parametrów. Długość okna FFT stanowi kompromis między rozdzielczością częstotliwościową a czasową - zazwyczaj stosuje się okna 2-4 s dla chodu. Nakładanie (ang. *overlap*) między kolejnymi oknami na poziomie 50-75 % dla STFT zapewnia płynną analizę. Wybór rodziny falki dla ciągłej transformaty falkowej (ang. *Continuous Wavelet Transform, CWT*) zależy od charakteru sygnału - falki Gaussa (gaus2) są optymalne dla sygnałów gładkich, podczas gdy Daubechies dla sygnałów o ostrych przejściach. Wstępne przetwarzanie sygnału (ang. *pre-processing*) obejmuje usuwanie trendu oraz zastosowanie okien wagowych (Hanna lub Hamminga) w celu redukcji przecieków spektralnych.

## 2.3 Metody zaawansowane

### 2.3.1 Aktualizacja zerowej prędkości (ang. *Zero Velocity Update*)

Algorytmy aktualizacji zerowej prędkości (ang. *Zero Velocity Update, ZUPT*) wykorzystują fundamentalną właściwość chodu ludzkiego - w trakcie każdego kroku stopa ma krótki moment, w którym jej prędkość względem podłożu wynosi zero. Ten moment podczas fazy podporu (ang. *stance phase*) stanowi naturalny punkt kalibracji pozwalający na korekcję dryfu czujników inercyjnych, który rośnie wraz z czasem trwania eksperymentu.

Podczas fazy podporu prawdziwa prędkość czujnika wynosi  $v_{true} = 0$ , jednak całkowanie pomiarów z akcelerometru prowadzi do estymowanej prędkości  $v_{est} \neq 0$  ze względu na bias akcelerometru, szum pomiarowy i błędy w estymacji orientacji. ZUPT koryguje ten błąd przez wymuszenie  $v_{est} = 0$  podczas wykrytych okresów zerowej prędkości, realizowane przez aktualizację filtru Kalmana.

Zastosowanie filtru komplementarnego opartego na kwaternionach z detekcją zerowej prędkości osiąga wysoką dokładność i błąd na poziomie  $< 0.37\%$  całkowitej przebytej odległości [16], co odpowiada  $\approx 0.3\text{ m}$  błędu dla trajektorii  $80\text{ m}$ . Jest to wynik o rząd wielkości lepszy niż metody bez korekcji ZUPT.

Najnowsze badania przedstawiły adaptacyjną detekcję ZUPT opartą na częstotliwości z automatycznym doborem progów w zależności od stanu ruchu [47]. Podejście *double zero-velocity detection* (podwójne wykrywanie zerowej prędkości, ZVD) z analizą długości interwałów stanów (ang. *Length of State Intervals*) przewyższa tradycyjne metody ze stałym progiem, redukując liczbę fałszywych detekcji o  $\approx 40\%$ . Adaptacyjny próg dostosowuje się automatycznie do różnych wzorców chodu: wolny chód charakteryzuje się dłuższymi fazami podporu wymagającymi niższych progów, podczas gdy bieg ma krótsze

fazy wymagające bardziej agresywnej detekcji.

Odporność na różne typy chodu (chodzenie, bieganie, wchodzenie po schodach) można osiągnąć poprzez zastosowanie wielu metod detekcji (ang. *multiple detectors*) [8]. Pierwsza metoda (ZVD1) dedykowana dla chodzenia stosuje tradycyjne progi na wielkość przyspieszenia i prędkości kątowej, podczas gdy druga metoda (ZVD2) optymalizowana dla biegania uwzględnia krótsze fazy podporu i wyższe amplitudy. Takie podejście osiąga porównywalną dokładność dla obu aktywności, stanowiąc przewagę nad algorytmami dostrojonymi tylko do jednego typu ruchu.

Ograniczenia metody ZUPT obejmują wymaganie montażu czujnika na stopie lub kostce (nie działa dla nadgarstka), trudności z bieganiem ze względu na krótsze fazy podporu, oraz błąd żyroskopu rosnący z czasem trwania eksperymentu, pozostający dominującym źródłem błędu długoterminowego.

### 2.3.2 Progi adaptacyjne

Progi adaptacyjne stanowią ewolucję prostych metod progowych. Pozwalają one na dynamiczne dostosowywanie parametrów detekcji do bieżących warunków i charakterystyki chodu użytkownika.

Metoda adaptacyjnego progu dla detekcji *heel strike* (HS) wykorzystuje model regresji liniowej oparty na trzech sygnałach z czujnika IMU umieszczonego na udzie: kącie uda  $\theta_z$ , prędkości kątowej  $\omega_z$  oraz przyspieszeniu osiowym  $a_x$  [51]. Próg kątowy dla detekcji HS jest obliczany adaptacyjnie w czasie rzeczywistym na podstawie wartości sygnałów w momencie osiągnięcia szczytu kąta uda:

$$\theta_h = B + W \cdot \begin{bmatrix} a_p \\ \theta_p \\ \omega_p \end{bmatrix}, \quad (2.6)$$

gdzie  $\theta_h$  jest progiem kątowym dla detekcji HS,  $a_p$ ,  $\theta_p$ ,  $\omega_p$  oznaczają odpowiednio przyspieszenie, kąt i prędkość kątową uda w momencie szczytu,  $B$  jest stałą, a  $W$  wektorem współczynników regresji. Dla detekcji *toe off* (TO) zastosowano stałe progi obserwowalne:  $\theta^* = 0^\circ$ ,  $\omega^* = 0.2 \frac{\text{rad}}{\text{s}}$  oraz  $a^* = 0 \frac{\text{rad}}{\text{s}^2}$ . Algorytm został wytrenowany na danych od 20 osób zdrowych oraz 6 pacjentów z hemiplegią, chodzących przy częstotliwościach kroków w zakresie 0.25-0.40 Hz. Osiągnięte błędy czasowe wyniosły dla HS:  $1.3 \pm 1.1\%$  cyklu chodu (osoby zdrowe) oraz  $1.2 \pm 0.7\%$  (pacjenci hemiplegiczni), a dla TO:  $1.8 \pm 1.1\%$  cyklu chodu (osoby zdrowe) oraz  $1.3 \pm 0.7\%$  (pacjenci hemiplegiczni<sup>8</sup>). Kluczowym osiągnięciem jest 100 % wskaźnik detekcji - algorytm nigdy nie pomija zdarzeń chodu, nawet przy znacznej asymetrii charakterystycznej dla pacjentów neurologicznych.

---

<sup>8</sup>Hemiplegia - połowiczne porażenie ciała, dotyczące jednej strony (prawej lub lewej) obejmujące jednocześnie rękę i nogę.

Alternatywne podejście stanowią adaptacyjne oscylatory (ang. *Adaptive Frequency Oscillators, AFO*), które oferują szczególnie interesującą możliwość detekcji z czujnika montowanego na głowie [44]. System wykorzystujący pulę czterech oscylatorów automatycznie dostosowuje się do kadencji chodu użytkownika w ciągu około  $\approx 7$  kroków. Montaż czujnika na głowie jest atrakcyjny w kontekście urządzeń noszonych naturalnie przez użytkowników, takich jak okulary czy słuchawki. Stabilizacja głowy podczas chodu, będąca naturalnym mechanizmem ludzkiego układu równowagi, zapewnia relatywnie czysty sygnał pomiarowy.

Charakterystyczną cechą metod adaptacyjnych jest wykorzystanie ruchomego okna (ang. *sliding window*) do analizy historii sygnału i dostosowania parametrów. Typowa długość okna wynosi od 5 do 15 kroków (10-30 s), co zapewnia balans między responsywnością na zmiany a stabilnością parametrów. Krótsze okna reagują szybciej na zmiany prędkości, ale są bardziej podatne na wartości odstające (ang. *outliers*), podczas gdy dłuższe okna są stabilniejsze, ale wolniej adaptują się do zmian warunków.

### 2.3.3 Dopasowywanie wzorca (ang. *Template matching*)

Dopasowywanie wzorca stanowi metodę bazującą na porównaniu badanego sygnału z biblioteką wzorców (ang. *templates*) reprezentujących typowe kroki. Największe badanie walidacyjne tej metody obejmowało 1 020 nagrani, 40 465 kroków i 230 uczestników [34]. Populacja badawcza była szczególnie zróżnicowana: 52 osoby zdrowe, 53 ortopedyczne i 125 neurologiczne, co zapewnia reprezentatywność wyników dla heterogenicznej populacji. Czujnik XSens MTw IMU montowany na stopie (częstotliwość próbkowania 100 Hz) dostarczał dane wykorzystywane przez algorytm operujący na bibliotece składającej się z 55 wzorców.

Metoda dopasowywania wzorca osiągnęła czułość (ang. *recall*)  $98.34\% \pm 3.00\%$  i precyzyję (ang. *precision*)  $98.30\% \pm 3.25\%$ , co daje F1-score równy 98.32 %. Dla porównania, zoptymalizowana metoda Pan-Tompkins osiągnęła *recall*  $97.82\% \pm 3.07\%$  i *precision*  $95.72\% \pm 4.56\%$  ( $F1 = 96.76\%$ ). Algorytm wykazał wysoką dokładność niezależnie od typu uczestnika - zarówno osoby zdrowe (*recall* 99.31 %), jak i pacjenci ortopedyczni (97.64 %) oraz neurologiczni (98.23 %) osiągnęli porównywalne wyniki. Największe wyzwanie stanowią niestandardowe wzorce ruchu: podczas gdy normalne kroki są wykrywane z *recall* 99.58 %, obroty U-turn osiągają tylko 83.87 % ze względu na nieregularną dynamikę ruchu.

Aby zrozumieć działanie metody, warto przeanalizować jej poszczególne etapy. Najpierw z okna sygnału ekstrahowane są cechy (zazwyczaj wielkość przyspieszenia i prędkość kątowa), następnie przeprowadzana jest normalizacja amplitudy i czasu - często z wykorzystaniem Dynamic Time Warping dla elastycznego dopasowania. W kolejnym kroku obliczana jest miara podobieństwa (ang. *correlation, Euclidean distance, DTW distance*)

dla każdego wzorca w bibliotece. Na końcu wybierany jest najlepiej pasujący wzorzec (ang. *best-matching template*) i krok jest wykrywany, jeśli podobieństwo przekracza zadany próg.

Liczba wzorców w bibliotece stanowi kompromis między dokładnością a złożonością obliczeniową. Biblioteka 55 wzorców używana w cytowanym badaniu zapewnia dobrą reprezentację różnorodności wzorców chodu, choć niektóre implementacje używają mniejszych bibliotek (10-20 wzorców) dla aplikacji działających w czasie rzeczywistym na urządzeniach mobilnych.

## 2.4 Głębokie uczenie w detekcji kroków

### 2.4.1 Konwolucyjne sieci neuronowe

Konwolucyjne sieci neuronowe (ang. *Convolutional Neural Networks*, CNN) wykazują szczególną skuteczność w automatycznej ekstrakcji cech z surowych sygnałów. Przegląd architektur 1D-CNN stosowanych w detekcji aktywności wykazuje typową strukturę obejmującą 5-9 warstw, z czego 2-3 stanowią warstwy konwolucyjne wykorzystujące kernele o rozmiarach  $1 \times 3-1 \times 5$  [52]. Architektura uzupełniona jest warstwami max-pooling dla redukcji wymiarowości oraz 1-2 warstwami w pełni połączonymi. W przypadku wielu czujników IMU stosuje się podejście multi-branch z osobnymi kernelami dla każdego sensora i późną fuzją sygnałów.

CNN z mechanizmami uwagi (ang. *attention mechanisms*) osiągają około 7 % poprawę dokładności przez adaptacyjne skupienie na istotnych oknach czasowych [52]. Lightweight CNN z podwójną uwagą (kanałową i czasową) przewyższa standardowe CNN na popularnych zbiorach danych (ang. *dataset*). Uwaga kanałowa pozwala modelowi nauczyć się, które osie sensora ( $x$ ,  $y$ ,  $z$ ) są najbardziej informatywne dla danej aktywności, podczas gdy uwaga czasowa identyfikuje kluczowe momenty w sekwencji.

Wariantem architektur konwolucyjnych przystosowanym do analizy danych sekwencyjnych są sieci TCN (ang. *Temporal Convolutional Networks*). Anderson i in. [3] wykorzystali TCN do segmentacji kroków, badając wpływ ilości danych treningowych, lokalizacji sensora oraz złożoności zadania ruchowego. Kluczowym odkryciem jest silna zależność wydajności od punktu mocowania czujnika oraz typu wykonywanego ruchu. Sensory montowane na stopie utrzymują wysoką dokładność zarówno podczas prostego chodu, jak i skrętów czy złożonych ruchów przejściowych. Z kolei sensory nadgarstkowe wykazują znaczną degradację wydajności wraz ze wzrostem złożoności ruchu.

Praktyczna implementacja wymaga uwzględnienia szeregu aspektów technicznych. Stabilność treningu zapewnia zastosowanie normalizacji wsadowej (ang. *batch normalization*) po każdej warstwie konwolucyjnej. Regularyzacja poprzez dropout (0.2-0.5) pomaga uniknąć przeuczenia modelu. Augmentacja danych obejmuje techniki takie jak zakrzywienie

czasowe (ang. *time warping*), zakrzywienie amplitudy (ang. *magnitude warping*), rotacje oraz permutacje. Typowo stosowane są okna wejściowe o stałej długości, zazwyczaj 2.5-5 s przy częstotliwości 50 lub 100 Hz.

Możemy zaobserwować, że algorytmy dla czujników dolnej kończyny mogą być trenowane na niewielkich datasetach (5-10 uczestników) osiągając wysoką wydajność, podczas gdy czujniki nadgarstkowe wymagają znacznie większej ilości danych treningowych dla porównywalnej dokładności [3].

#### 2.4.2 Rekurencyjne sieci neuronowe

Rekurencyjne sieci neuronowe (ang. *Recurrent Neural Networks*, RNN), w szczególności architektury LSTM (ang. *Long Short-Term Memory*) oraz GRU (ang. *Gated Recurrent Units*), naturalnie modelują sekwencje czasowe dzięki wbudowanemu mechanizmowi pamięci. Zdolność do zachowania informacji z wcześniejszych kroków czyni je szczególnie przydatnymi w analizie wzorców chodu, gdzie kolejne fazy cyklu są ze sobą ścisłe powiązane.

Sarshar i in. [38] zastosowali głęboką architekturę LSTM do estymacji fazy chodu, osiągając wysoką dokładność na zbiorze testowym obejmującym 64 niezależnych uczestników. System wykorzystywał czujniki zamontowane na kostkach, mierzące prędkość kątową, przyspieszenie oraz orientację w oknie czasowym 2.51 s. Kluczową zaletą jest zdolność do wykrywania wszystkich kroków, włącznie z inicjacją chodu, oraz odporność na szum i nieregularności chodu. Ograniczeniem pozostaje jednak niższa wydajność podczas wykonywania skrętów.

Modyfikacja standardowej architektury LSTM może znacząco poprawić wykrywanie zdarzeń cyklu chodu. Tan i in. [43] wykazali, że zmodyfikowany LSTM przewyższa klasyczne algorytmy state-of-the-art w detekcji momentu kontaktu pięty (HS) i oderwania palców od podłoga (TO). Znaczący spadek wydajności w warunkach zewnętrznych (szczególnie dla detekcji toe-off) wskazuje na istotny wpływ nieregularnego terenu i zmiennych warunków środowiskowych na dokładność algorytmów.

Architektura hybrydowa łącząca zalety CNN i RNN oferuje dodatkowe korzyści. Shi i in. [39] zaproponowali Sequential Convolution LSTM (SConvLSTM), w której warstwy konwolucyjne ekstrahują lokalne cechy czasowe, a następnie dwukierunkowy LSTM (ang. *Bidirectional LSTM*) modeluje zależności w obu kierunkach czasu. Podejście to unika błędów segmentacji poprzez wykorzystanie ramek o stałej długości oraz uwzględnienie kontekstu z przyszłych ramek przy klasyfikacji bieżącej aktywności. Model osiągnął wysoką wydajność na popularnych benchmarkach HAR.

Kluczowym parametrem projektowym w architekturach RNN jest wybór długości okna czasowego. Krótkie okna (poniżej 1 s) zapewniają szybką reakcję, lecz nie obejmują pełnego cyklu chodu. Średnie okna (1-3 s) stanowią optymalny kompromis dla większości

zastosowań, łącząc odpowiedni kontekst z akceptowalnym opóźnieniem. Długie okna (powyżej 3 s) oferują lepszy kontekst całego cyklu chodu, ale zwiększą opóźnienie oraz wymagania pamięciowe systemu.

Praktyczne wdrożenie wymaga rozważenia kompromisu między dokładnością predykcji a czasem reakcji systemu, co jest szczególnie istotne w aplikacjach czasu rzeczywistego, takich jak sterowanie protezami czy systemy wspomagające chód.

### 2.4.3 Transformery

Architektury Transformer, pierwotnie opracowane dla przetwarzania języka naturalnego, wprowadzają mechanizm *self-attention*, który umożliwia modelowanie długoterminowych zależności w sekwencjach bez ograniczeń charakterystycznych dla sieci rekurencyjnych. W przeciwieństwie do LSTM, który przetwarza dane sekwencyjnie, mechanizm uwagi pozwala na równoległe przetwarzanie całej sekwencji i bezpośrednie modelowanie zależności między dowolnie odległymi krokami czasowymi.

W kontekście analizy chodu transformery znajdują zastosowanie w różnorodnych zadaniach. Lu i in. [30] zaproponowali GaitFormer, dwustrumieniową architekturę wykorzystującą mechanizm *multihead autocorrelation* oraz wieloskalowe warstwy konwolucyjne do rozpoznawania tożsamości na podstawie wzorców chodu. System osiągnął wyniki state-of-the-art na benchmarkowych datasetach w kontekście zastosowań dla Przemysłu 4.0.

Ji i in. [24] zastosowali transformery do estymacji fazy cyklu chodu, łącząc warstwy konwolucyjne temporalne z warstwami transformerów. Kluczową zaletą tego podejścia jest niejawne modelowanie (ang. *implicit modeling*) - model nauczył się reprezentacji fazy bez wymuszania konkretnej struktury cyklicznej, co zapewnia doskonałą generalizację zarówno na stabilnym terenie, jak i podczas przejścia między różnymi typami nawierzchni. Stanowi to przewagę nad tradycyjnymi metodami wymagającymi jawnego modelowania cyklu.

W dalszych pracach ten sam zespół [24] rozszerzył zastosowanie transformerów na predykcję pełnej kinematyki i kinetyki stawów. Opracowana architektura wykorzystuje oddzielne sieci transformerów dla kinematyki i kinetyki, przetwarzając sygnały sEMG (elektromiografia powierzchniowa, ang. *surface electromyography*<sup>9</sup>) oraz IMU jako wejście. System umożliwia predykcję długohoryzontową (ang. *long-horizon prediction*), co jest kluczowe dla sterowania w systemach noszonych i robotycznych.

Głównym wyzwaniem w stosowaniu transformers jest wysoka złożoność obliczeniowa mechanizmu uwagi, rosnąca kwadratowo względem długości sekwencji wejściowej ( $O(n^2)$ , gdzie  $n$  to liczba kroków czasowych), co ogranicza możliwość przetwarzania bardzo długich sekwencji na urządzeniach o ograniczonych zasobach. Praktyczne implementacje często wykorzystują hybrydowe architektury łączące CNN z transformerami - warstwy kon-

---

<sup>9</sup>sEMG - nieinwazyjna metoda do rejestracji, analizy i interpretacji elektrycznej aktywności mięśni szkieletowych, wykorzystująca elektrody umieszczone na skórze.

wolucyjne ekstrahują cechy i redukują wymiarowość, podczas gdy transformer modeluje globalne zależności temporalne w zredukowanej przestrzeni cech.

#### **2.4.4 Uczenie transferowe (ang. *Transfer Learning*) i uczenie maszynowe na urządzeniach brzegowych**

Praktyczne wdrożenie systemów rozpoznawania aktywności na urządzeniach noszonych, takich jak zegarki sportowe czy smartfony, wymaga rozwiązania dwóch kluczowych wyzwań: ograniczonej dostępności danych treningowych oraz ograniczonych zasobów obliczeniowych urządzeń. *Transfer Learning* oraz TinyML (ang. *Tiny Machine Learning*) stanowią komplementarne podejścia rozwiązuające te problemy.

##### **Uczenie transferowe i personalizacja modeli**

Uczenie transferowe (ang. *Transfer Learning*) umożliwia wykorzystanie wiedzy nabyciej podczas treningu na dużych zbiorach do nowych zastosowań z ograniczoną ilością danych. Luu i in. [31] wykazali, że model wytrenowany na danych z jednego typu urządzenia może być skutecznie zaadaptowany do innego urządzenia przy użyciu minimalnej ilości danych adaptacyjnych (około 30 sekund na użytkownika), osiągając wysoką dokładność liczenia kroków. Istotnym odkryciem jest możliwość wykorzystania wyłącznie akcelerometru zamiast pełnego IMU, co znaczco redukuje koszty sprzętowe i zużycie energii przy zachowaniu wysokiej wydajności.

Autoenkodery znajdują zastosowanie zarówno w preprocesingu sygnału, jak i detekcji anomalii. Denoising Autoencoder może znaczco poprawić stosunek sygnału do szumu (ang. *signal-to-noise ratio, SNR*) w surowych danych, co przekłada się na lepszą wydajność klasyfikatorów. W kontekście klinicznym, autoenkodery umożliwiają wykrywanie anomalii we wzorcach chodu poprzez uczenie się normalnych wzorców ruchu i identyfikację znaczących odchyleń, co jest szczególnie przydatne w monitorowaniu objawów neurologicznych takich jak *freezing of gait* w chorobie Parkinsona.

##### **TinyML - uczenie maszynowe na mikrokontrolerach**

TinyML reprezentuje nowy paradymat umożliwiający wdrożenie modeli uczenia maszynowego bezpośrednio na mikrokontrolerach o ograniczonych zasobach [48]. Kluczowe techniki obejmują kwantyzację (redukcję precyzji wag) oraz przycinanie (ang. *pruning*) sieci neuronowych w celu uzyskania modeli o rozmiarze poniżej 100 *KB*, które mogą działać na urządzeniach zaledwie kilkudziesięcioma kilobajtami pamięci RAM (zazwyczaj 32-64 *KB*).

Podejście TinyML oferuje szereg istotnych korzyści. Przetwarzanie odbywa się w czasie rzeczywistym z opóźnieniem poniżej 10 *ms*, przy jednoczesnej niezależności od łącz-

ności sieciowej i przetwarzania w chmurze. Dane pozostają na urządzeniu, co zwiększa prywatność użytkownika. Zużycie energii jest drastycznie niższe w porównaniu do przesyłania danych przez internet. Dodatkowo, możliwe jest wdrożenie treningu przyrostowego (ang. *continual learning*) bezpośrednio na urządzeniu, co pozwala na adaptację modelu do indywidualnych wzorców użytkownika.

Praktyczne zastosowania TinyML w rozpoznawaniu aktywności obejmują wykrywanie upadków na urządzeniach o zasobach rzędu kilkudziesięciu kilobajtów, rozpoznawanie gestów na urządzeniach noszonych bez połączenia z internetem, oraz klasyfikację aktywności w systemach IoT. Rozwój dedykowanych akceleratorów sprzętowych, takich jak *ARM Cortex-M55* z *Ethos-U55*, dodatkowo zwiększa możliwości przetwarzania na urządzeniach końcowych. Umożliwia to wdrażanie bardziej zaawansowanych algorytmów przy zachowaniu niskiego zużycia energii.

## 2.5 Wpływ lokalizacji na dokładność detekcji

Miejsce montażu czujnika IMU ma fundamentalny wpływ na dokładność detekcji kroków i rozpoznawania aktywności, z różnicami sięgającymi nawet kilkudziesięciu punktów procentowych między optymalnym a nieodpowiednim umiejscowieniem. Wybór lokalizacji stanowi kompromis między dokładnością pomiarów, praktycznością użytkowania oraz komfortem użytkowników. Zrozumienie tych zależności jest kluczowe zarówno dla projektowania systemów pomiarowych, jak i interpretacji wyników badań wykorzystujących różne urządzenia.

### 2.5.1 Hierarchia dokładności według lokalizacji czujnika

Systematyczne badania porównawcze wykazują wyraźną hierarchię dokładności w zależności od lokalizacji montażu [15, 35, 41]. Czujniki montowane na kostce lub stopie osiągają najwyższą precyzję we wszystkich warunkach, z błędami poniżej 3 % dla typowego chodu. Foster i in. [15] wykazali, że akcelerometr montowany na kostce utrzymuje dokładność bliską 100 % niezależnie od prędkości ruchu, podczas gdy urządzenia montowane na biodrze mogą wykazywać błędy przekraczające 70 % przy bardzo wolnym chodzie.

Montaż na udzie stanowi drugi najdokładniejszy wariant, oferując błędy poniżej 3 % dla normalnego chodu przy jednoczesnej większej praktyczności niż montaż na stopie. Czujniki montowane na talii wykazują umiarkowaną skuteczność (błędy 2-4 % dla normalnego chodu), co sprawia, że są dobrym kompromisem między precyzją a wygodą użytkowania. Montaż na nadgarstku oferuje najniższą dokładność dla chodu (błędy 3-9 %), przy czym wydajność znaczco poprawia się podczas biegu [35]. Górnego ramienia zajmuje pozycję pośrednią, z błędami 6-15 % wykazującymi silną zależność od prędkości ruchu [41].

## 2.5.2 Wpływ prędkości ruchu na dokładność

Prędkość ruchu istotnie wpływa na dokładność detekcji, przy czym efekt ten różni się znacząco w zależności od lokalizacji czujnika. Hicks i in. [20] wykazali, że podczas bardzo wolnego chodu (około  $1.5 \frac{km}{h}$ ) czujniki montowane na biodrze wykazują znaczne błędy w detekcji kroków, podczas gdy montaż na udzie utrzymuje wysoką precyzję. Wraz ze wzrostem prędkości do normalnego chodu ( $3-4.5 \frac{km}{h}$ ) błędy dla większości lokalizacji znacząco maleją. Przy biegu (około  $9 \frac{km}{h}$ ) obserwuje się odwrócenie tych zależności - czujniki montowane na biodrze osiągają wtedy bardzo wysoką dokładność, podczas gdy skuteczność czujników na udzie nieznacznie maleje.

Ten efekt ma szczególne znaczenie praktyczne w badaniach populacji o ograniczonej mobilności, takich jak osoby starsze czy pacjenci z chorobami neurologicznymi, którzy często poruszają się z prędkościami poniżej normalnych. W takich zastosowaniach montaż na dolnej kończynie jest krytyczny dla uzyskania wiarygodnych pomiarów.

## 2.5.3 Implikacje praktyczne i rekomendacje

W kontekście projektowania systemów rozpoznawania aktywności, wybór lokalizacji czujnika powinien uwzględniać specyfikę zastosowania:

- **Precyzyjne pomiary chodu** (rehabilitacja, badania kliniczne): montaż na kostce, stopie lub udzie zapewnia najwyższą dokładność niezależnie od prędkości ruchu.
- **Długoterminowy monitoring aktywności** (badania populacyjne): montaż na nadgarstku oferuje najlepszą akceptację przez użytkowników mimo niższej dokładności przy wolnym chodzie.
- **Populacje o ograniczonej mobilności**: zalecany jest montaż na dolnej kończynie ze względu na utrzymanie wysokiej dokładności przy niskich prędkościach ruchu.
- **Systemy wielomodalne**: kombinacja czujników w różnych lokalizacjach oferuje kompromis między dokładnością a praktycznością przy jednoczesnym zwiększeniu odporności na błędy pojedynczego sensora.

Należy podkreślić, że współczesne algorytmy uczenia maszynowego mogą częściowo kompensować nieoptimalne umiejscowienie czujnika poprzez uczenie się specyficznych wzorców sygnału dla danej lokalizacji. Jednak różnice w dokładności między lokalizacjami pozostają istotne i powinny być uwzględniane zarówno przy projektowaniu systemów, jak i interpretacji wyników badań porównujących urządzenia montowane w różnych miejscach.

## 2.6 Wyzwania i ograniczenia obecnych rozwiązań

### 2.6.1 Problemy dobrze rozwiązane

Analiza chodu w kontrolowanych warunkach osiągnęła dojrzałość techniczną, konsekwentnie zapewniając dokładność na poziomie 95-99 % dla standardowego chodu. Systemy montowane na stopie wykorzystujące aktualizacje zerowej prędkości (ang. *Zero Velocity Update*, ZUPT) osiągają skumulowany błąd pozycji poniżej 1 % długości trajektorii. Metody oparte na szablonach (ang. *template-based methods*) wykazują wysoką precyzję i czułość na heterogenicznych populacjach obejmujących zarówno osoby zdrowe, jak i pacjentów ortopedycznych oraz neurologicznych [34].

Fuzja danych sensorycznych stanowi obecnie technologię dojrzałą i dobrze zrozumianą. Łączenie sygnałów z akcelerometru i żyroskopu poprzez filtry komplementarne lub filtry Kalmana efektywnie równoważy kompromis między dryfem a szumem pomiarowym. Akcelerometr dostarcza długoterminowo stabilnej referencji orientacji względem grawitacji, podczas gdy żyroskop zapewnia precyzyjne krótkoterminowe pomiary rotacji. Integracja magnetometru dla estymacji kursu jest dobrze ugruntowana, choć wymaga uwzględnienia interferencji magnetycznych w środowiskach zamkniętych, takich jak konstrukcje stalowe w budynkach.

Niezawodność komercyjnych urządzeń noszonych osiągnęła poziom wystarczający dla zastosowań związanych z monitoringiem aktywności fizycznej. Popularne urządzenia konsumenckie wykazują błąd liczenia kroków na poziomie 2-5 % dla normalnego chodu. Algorytmy zostały zoptymalizowane pod kątem równowagi między czasem pracy na baterii a dokładnością. Przetwarzanie w czasie rzeczywistym na urządzeniach o ograniczonych zasobach działa efektywnie dzięki sprawdzonym implementacjom zarówno klasycznych algorytmów, jak i rozwiązań TinyML.

### 2.6.2 Otwarte wyzwania

Pomimo postępów w analizie chodu, szereg problemów ogranicza zastosowanie systemów noszonych w rzeczywistych warunkach.

#### Złożone wzorce ruchu

Nieregularne ruchy znaczająco obniżają dokładność detekcji. Skręty o  $180^\circ$  wykazują spadek czułości o około 16 punktów procentowych względem prostoliniowego chodu [34]. Ruchy wsteczne, inicjacja i zakończenie chodu osiągają niższą precyzję niż ustabilizowany chód. Dla nieregularnych wzorców nawet eksperci nie zgadzają się co do granic kroków, co pokazuje podstawowy problem w definiowaniu segmentacji dla patologicznych wzorców ruchu.

## Zmienne położenie urządzenia

Niekontrolowana orientacja czujnika jest problematyczna dla urządzeń noszonych swobodnie. Używanie smartfona podczas chodu (pisanie wiadomości) powoduje fałszywe detekcje, gdy ruchy rąk są błędnie interpretowane jako kroki. Urządzenie w kieszeni daje niestabilne wyniki ze względu na losową orientację. Najtrudniejszy jest scenariusz swobodnie trzymanego urządzenia. Aktywności z ograniczonym ruchem ramion (pchanie wózka) szczególnie wpływają na wydajność czujników nadgarstkowych.

## Czynniki środowiskowe i sensoryczne

Interferencje magnetyczne w pomieszczeniach zakłócają pomiary magnetometru, obniżając dokładność estymacji kierunku. Efekty temperaturowe wymagają kompensacji lub częstej kalibracji. Bardzo wolny chód (poniżej  $2 \frac{km}{h}$ ) jest szczególnie problematyczny - urządzenia nadgarstkowe wykazują błędy przekraczające 75 % [35]. Zmiany nawierzchni oraz przeszkody znaczaco zmieniają wzorce ruchu.

Częstotliwość próbkowania stanowi kompromis między czasem pracy baterii a precyzją pomiarów. Niska częstotliwość ( $10 \text{ Hz}$ ) pomija szybkie ruchy, podczas gdy wysoka (powyżej  $100 \text{ Hz}$ ) szybko wyczerpuje baterię. Wybór optymalnej wartości zależy od specyfiki zastosowania i wymaga indywidualnej optymalizacji dla każdego scenariusza użycia.

## Stabilność długoterminowa

Dryf i szum czujników MEMS powodują kumulację błędu w czasie. Błąd losowy prędkości kątowej stanowi główne źródło problemów w systemach z aktualizacjami zerowej prędkości [8]. Długoterminowa stabilność wymaga zewnętrznych źródeł referencyjnych, takich jak GPS czy punkty charakterystyczne w środowisku.

## Ograniczona walidacja populacji

Większość algorytmów jest rozwijana na zdrowych, młodych dorosłych, podczas gdy aplikacje obejmują znacznie szerszą populację. Brakuje walidacji na osobach z zaburzeniami chodu, takich jak pacjenci po udarze czy z chorobami neurologicznymi. Podobnie niedostatecznie zbadane są scenariusze z użytkownikami pomocy do chodzenia (laski, balkoniki) oraz skrajne grupy wiekowe - dzieci poniżej 10 lat i osoby powyżej 80 lat. Dodatkowo brakuje walidacji uwzględniającej różnorodność budowy ciała, w tym otyłość, amputacje czy znaczące różnice wzrostu. Ten brak reprezentatywności ogranicza zaufanie do systemów w rzeczywistych zastosowaniach medycznych.

### 2.6.3 Luki metodologiczne

Poza problemami technicznymi, istotne ograniczenia wynikają z praktyk badawczych i dostępności odpowiednich danych.

#### Problemy walidacji

Większość badań wykorzystuje dane z krótkich testów (spacery 10-30 m lub pomiary poniżej 1 minuty), co nie odzwierciedla precyzji w ciągu całego dnia. Testowane są często tylko typowe prędkości chodu, pomijając wolny chód i bieg. Dokładność w warunkach rzeczywistych jest zazwyczaj niższa niż raportowana w kontrolowanych warunkach laboratoryjnych. Rzadko rapportuje się wyniki z pełnego dnia - większość badań pokazuje rezultaty z wyselekcjonowanych segmentów danych, co może sztucznie zawyżać rzeczywistą wydajność systemów.

#### Ograniczenia zbiorów danych

Brak standardowych zbiorów testowych utrudnia rzetelne porównanie metod. Większość datasetów zawiera głównie zdrowych uczestników, z niewystarczającą reprezentacją patologicznych wzorców chodu. Ograniczona różnorodność dotyczy przede wszystkim wąskich zakresów wiekowych (głównie 20-50 lat), różnych kategorii BMI, użytkowników pomocy do chodzenia oraz osób z chorobami neurologicznymi. Brakuje długoterminowych badań śledzących tych samych użytkowników przez tygodnie lub miesiące, co uniemożliwia ocenę stabilności algorytmów w czasie.

## 2.7 Zastosowania praktyczne

### 2.7.1 Inteligentne zegarki i opaski sportowe

Komercyjne urządzenia pomiarowe stanowią najbardziej rozpowszechnione zastosowanie technologii rozpoznawania aktywności, osiągając rynek setek milionów użytkowników. Dokładność tych urządzeń różni się znaczco w zależności od producenta, modelu oraz warunków użytkowania.

Apple Watch wykorzystuje akcelerometr na nadgarstku do pomiaru ruchu ramienia. Badanie na 71 uczestnikach wykazało średni błąd 1.6% przy szybkim tempie chodu (4.0 mph) [46]. Kalibracja przez 20-minutowy spacer lub bieg poprawia dokładność poprzez uczenie długości kroku użytkownika [5]. Główne problemy to fałszywe detekcje kroków podczas jazdy samochodem czy pracy przy biurku oraz tendencja do przeszacowywania.

Urządzenia Fitbit noszone na nadgarstku analizują wzorce ruchu. Montaż na nadgarstku zwiększa podatność na fałszywe detekcje z ruchów rąk podczas codziennych

czynności. Przegląd 67 badań wykazał tendencję do przeszacowywania o 700-1800 kroków dziennie [13]. Nowsze badanie z 2024 roku potwierdziło średnie przeszacowanie o 1392-1887 kroków dziennie, zależnie od grupy wiekowej [9].

Garmin Forerunner, zoptymalizowany dla zastosowań sportowych, wykazuje średni błąd poniżej 5 % przy prędkościach  $3\text{-}5 \frac{\text{km}}{\text{h}}$  w warunkach kontrolowanych [32]. Urządzenia Garmin osiągają lepszą precyzję przy wyższych prędkościach biegu niż chodu dzięki optymalizacji algorytmów pod kątem sportowców.

Porównawcze testy konsumenckie pokazują, że urządzenia dedykowane do biegania (Garmin, COROS) osiągają lepszą dokładność niż zegarki codziennego użytku (ang. *lifestyle*) [21]. Ogólnie, wszystkie urządzenia nadgarstkowe wykazują wyższą skuteczność przy szybszym tempie w porównaniu do wolnego chodu i mogą generować fałszywe detekcje podczas aktywności bez chodu.

### 2.7.2 Nawigacja wewnętrzna

Nawigacja wewnętrzna (ang. *Pedestrian Dead Reckoning*, PDR) umożliwia lokalizację bez GPS poprzez integrację kolejnych kroków. System PDR składa się z czterech głównych komponentów: detekcji kroków, estymacji długości kroku, określenia kierunku ruchu oraz aktualizacji pozycji poprzez kumulację przemieszczeń [26].

Współczesne systemy PDR wykorzystują fuzję wielu czujników dla poprawy dokładności [14]. Połączenie IMU z magnetometrem i sygnałami BLE (ang. *Bluetooth Low Energy*) redukuje błąd poprzez okresowe korekty oparte na punktach referencyjnych. Typowy błąd względny wynosi około 7.4 % dla prostoliniowego chodu i wzrasta do około 12.3 % podczas zmian kierunku.

Wybór lokalizacji czujnika znacząco wpływa na dokładność systemu. Montaż na stopie zapewnia najwyższą precyzję dzięki możliwości zastosowania aktualizacji zerowej prędkości, osiągając skumulowany błąd poniżej 1 %. Smartfon trzymany w ręku jest najbardziej praktyczny, lecz podatny na błędy przekraczające 3 000 kroków dziennie. Montaż w tali oferuje dobrą równowagę między dokładnością a praktycznością, podczas gdy montaż na nadgarstku, mimo wygody, cierpi na wpływ swobodnych ruchów ramienia. Montaż na głowie stanowi podejście rozwijane, wykorzystujące naturalną stabilizację głowy podczas chodu.

Główne wyzwania obejmują obsługę skrętów, które powodują spadek czułości o około 16 punktów procentowych względem prostoliniowego chodu. Zmienne prędkości ruchu wymagają adaptacyjnych parametrów algorytmów. Różnorodne sposoby trzymania urządzenia (pisanie wiadomości, rozmawianie, swobodne ruchy w kieszeni) wprowadzają dodatkową niepewność. Interferencje magnetyczne w środowiskach zamkniętych stanowią trwałe ograniczenie dla estymacji kierunku ruchu.

### 2.7.3 Monitoring medyczny i rehabilitacja

Systemy rozpoznawania aktywności znajdują zastosowanie w monitoringu chorób neurologicznych i rehabilitacji ortopedycznej. W chorobie Parkinsona długość kroku silnie koreluje z zaawansowaniem choroby, umożliwiając obiektywny monitoring progresji [36]. Rehabilitacja poudarowa wykorzystuje automatyczną detekcję parametrów chodu dla oceny postępów pacjenta.

Walidacja metod opartych na szablonach w badaniu na 230 uczestnikach obejmujących osoby zdrowe, pacjentów ortopedycznych i neurologicznych wykazała wysoką dokładność we wszystkich grupach, z czułością i precyzją przekraczającą 97% [34]. Błędy czasowe detekcji początku i końca kroku wynoszą zaledwie 0.04-0.07 s, co jest wystarczające dla zastosowań klinicznych.

Kluczowe zastosowania obejmują wykrywanie upadków u osób starszych, ciągły monitoring parametrów chodu jako wyznaczników wczesnej diagnozy oraz ocenę skuteczności interwencji terapeutycznych. Systemy oparte na czujnikach inercyjnych oferują przewagę nad oceną laboratoryjną poprzez możliwość długoterminowego monitoringu w warunkach naturalnych, co dostarcza bardziej reprezentatywny obraz funkcjonowania pacjenta.

## 2.8 Kierunki rozwoju i przyszłe badania

Rozwój systemów rozpoznawania aktywności koncentruje się na kilku kluczowych obszarach. Uczenie maszynowe na urządzeniach brzegowych (TinyML) umożliwia przetwarzanie w czasie rzeczywistym bez połączenia z chmurą, przy zużyciu energii poniżej 1 mW [48]. Uczenie federacyjne<sup>10</sup> oferuje trening rozproszony z zachowaniem prywatności danych, podczas gdy uczenie samo-nadzorowane redukuje potrzebę ręcznej anotacji danych.

Zaawansowana fuzja czujników łączy dane z wielu źródeł (IMU, kamera, czujniki ciśnienia, EMG) dla zwiększenia dokładności i odporności na błędy. Adaptacyjne podejście dynamicznie dostosowują wagę czujników w zależności od kontekstu (środowisko wewnętrzne/zewnętrzne, typ aktywności, sposób noszenia urządzenia).

Przetwarzanie uwzględniające kontekst automatycznie dostosowuje parametry algorytmów do warunków użytkowania, poprawiając dokładność przy zmiennych scenariuszach. Dedykowane akceleratory sprzętowe w nowoczesnych mikrokontrolerach umożliwiają coraz bardziej zaawansowane algorytmy przy utrzymaniu niskiego zużycia energii.

Kluczowe potrzeby badawcze obejmują cztery główne obszary. Po pierwsze, **standardyzacja** wymaga opracowania ujednoliconych metryk i protokołów testowych, publicznych zbiorów referencyjnych oraz reprodukowalnych algorytmów dla rzetelnego porównania me-

<sup>10</sup>ang. *Federated Learning* - Technika uczenia maszynowego, która polega na trenowaniu modelu AI na zdecentralizowanych danych przechowywanych na wielu urządzeniach [25]

tod. Po drugie, **walidacja w warunkach rzeczywistych** powinna obejmować długoterminowe badania z różnorodnymi aktywnościami i warunkami użytkowania, wykraczające poza kontrolowane testy laboratoryjne. Po trzecie, **adaptacja kliniczna** wymaga określenia ścieżek certyfikacji medycznej, wymagań dokładności na poziomie medycznym oraz integracji z systemami dokumentacji medycznej. Po czwarte, **personalizacja** wymaga rozwoju metod indywidualnej kalibracji i uczenia adaptacyjnego, uwzględniających wiek, stan zdrowia oraz pomoce do chodzenia pacjentów.

Realizacja tych kierunków rozwoju wymaga interdyscyplinarnej współpracy między inżynierią, medycyną i naukami o danych, ze szczególnym uwzględnieniem aspektów etycznych i ochrony prywatności użytkowników.

## 2.9 Podsumowanie

Analiza literatury ujawnia znaczącą dojrzałość dziedziny detekcji kroków z czujników IMU. Ewolucja od prostych metod progowych do współczesnych rozwiązań uczenia głębokiego przyniosła wzrost dokładności z około 85-90 % do 95-99 %. Metody oparte na szablonach i sieciach neuronowych osiągają obecnie precyzję porównywalną z metodami referencyjnymi w warunkach klinicznych [34, 38].

Wydajność systemu silnie zależy od lokalizacji czujnika. Montaż na kostce lub stopie zapewnia najwyższą dokładność, następnie udo, talia i nadgarstek [15, 35]. Wybór lokalizacji stanowi kompromis między dokładnością a wygodą użytkowania, przy czym różnice mogą przekraczać 50 punktów procentowych.

Największe wyzwania pozostają w warunkach rzeczywistych: nieregularne wzorce ruchu (skręty, ruchy wsteczne), zmienna orientacja urządzenia (pisanie, rozmowa) oraz bardzo wolny chód. Skuteczność w warunkach codziennych jest zazwyczaj niższa niż w laboratorium, co wskazuje na potrzebę dalszych badań nad odpornością algorytmów.

Przyszłość dziedziny koncentruje się na czterech głównych kierunkach: przetwarzaniu lokalnym na urządzeniach o ograniczonych zasobach, algorytmach dostosowujących się do kontekstu, fuzji wielu typów czujników oraz wprowadzeniu do zastosowań klinicznych. Ten ostatni obszar wymaga standaryzacji i długoterminowej walidacji systemów w warunkach rzeczywistych.



# Rozdział 3

## Przedmiot pracy

Niniejszy rozdział przedstawia szczegółowy opis zaproponowanego rozwiązania problemu estymacji kroków z czujników inercyjnych. W odróżnieniu od istniejących prac, które koncentrują się zazwyczaj na pojedynczej metodzie detekcji lub ograniczonych scenariuszach testowych, niniejsza praca proponuje kompleksowe podejście łączące budowę dedykowanego stanowiska pomiarowego z implementacją i obiektywnym porównaniem pięciu fundamentalnie różnych algorytmów detekcji. Kluczowym aspektem jest utworzenie ekosystemu narzędzi umożliwiających nie tylko zbieranie danych i ich analizę w czasie rzeczywistym, ale także weryfikację algorytmów na danych zewnętrznych źródeł poprzez dedykowany moduł interpretera.

### 3.1 Koncepcja rozwiązania

Proponowane rozwiązanie opiera się na trzech głównych elementach: sprzętowym stanowisku pomiarowym, oprogramowaniu analitycznym oraz metodologii walidacji. Ta architektura została zaprojektowana z myślą o zapewnieniu pełnej kontroli nad procesem pomiarowym, elastyczności w doborze parametrów detekcji oraz możliwości obiektywnego porównania różnych podejść algorytmicznych.

#### 3.1.1 Architektura systemu

System składa się z następujących komponentów funkcjonalnych:

1. **Warstwa sprzętowa** - Stanowisko pomiarowe oparte na mikrokontrolerze Raspberry Pi Pico 2W wyposażone w dwa moduły czujników inercyjnych (Wave-share Pico-10DOF-IMU z układem MPU9250 oraz Adafruit ST 9-DoF z układami LSM6DSOX i LIS3MDL), komunikujące się z komputerem poprzez sieć Wi-Fi.
2. **Warstwa oprogramowania pomiarowego** - Oprogramowanie (ang. *firmware*) mikrokontrolera obsługujące komunikację z sensorami poprzez magistralę I<sup>2</sup>C, im-

plementujące protokoły UDP i HTTP do transmisji danych oraz zarządzające punktem dostępowym Wi-Fi.

3. **Warstwa analizy w czasie rzeczywistym** - Aplikacja z interfejsem graficznym umożliwiająca wizualizację surowych danych, manualną anotację kroków (wyznaczenie wartości odniesienia, ang. *ground truth*) oraz uruchamianie algorytmów detekcji z natychmiastową prezentacją wyników.
4. **Warstwa analizy wsadowej** - Moduł do automatycznej analizy wielu nagrań bez konieczności interakcji użytkownika, generujący raporty porównawcze dla wszystkich zaimplementowanych algorytmów.
5. **Warstwa integracji danych** - Interpreter umożliwiający konwersję danych z zewnętrznych badań (szczególnie testów TUG otrzymanych od promotora pracy) do formatu wykorzystywanego przez system analityczny.

Taka modularna struktura pozwala na niezależny rozwój poszczególnych komponentów oraz łatwą rozbudowę systemu o dodatkowe algorytmy czy metody oceny jakości detekcji.

### 3.1.2 Przyjęte założenia projektowe

W projektowaniu systemu kierowano się następującymi założeniami:

- **Uniwersalność parametrów** - Zaimplementowane algorytmy powinny działać z jednolitymi parametrami niezależnie od punktu montażu sensora (kieszeń, nadgarstek, kostka), eliminując potrzebę ręcznej kalibracji dla każdego przypadku użycia.
- **Reprodukwalność** - Wszystkie eksperymenty muszą być możliwe do powtórzenia przez innych badaczy, co wymaga szczegółowej dokumentacji parametrów, protokołów testowych oraz dostępu do kodu źródłowego.
- **Transparentność oceny** - System powinien dostarczać wielowymiarowej oceny jakości detekcji (precyza, czułość, F1-score, błąd liczby kroków, czas wykonania), umożliwiając identyfikację mocnych i słabych stron każdej metody.
- **Efektywność obliczeniowa** - Algorytmy muszą być na tyle wydajne, aby umożliwiać szybkie przetwarzanie na komputerach PC, z perspektywą przyszłej implementacji na urządzeniach wbudowanych.
- **Dostępność technologiczna** - Wykorzystanie powszechnie dostępnych, niedrogich komponentów sprzętowych oraz otwartych narzędzi programistycznych zapewnia niski próg wejścia dla potencjalnych badaczy kontynuujących ten kierunek.

## 3.2 Wybór i uzasadnienie metod algorytmicznych

Dobór algorytmów detekcji kroków był wynikiem analizy literatury przedstawionej w rozdziale 2 oraz dążenia do reprezentacji głównych nurtów metodologicznych. Zdecydowano się na implementację pięciu fundamentalnie różnych podejść, które łącznie pokrywają spektrum od prostych metod klasycznych po zaawansowane techniki hybrydowe.

### 3.2.1 Detekcja szczytów (ang. *Peak Detection*)

**Uzasadnienie wyboru:** Metoda detekcji szczytów stanowi jedno z najprostszych i najbardziej intuicyjnych podejść do identyfikacji kroków. Jest szeroko stosowana w urządzeniach komercyjnych ze względu na niską złożoność obliczeniową i łatwość implementacji. Jej włączenie do zestawu testowego pozwala na ustalenie bazowej linii wydajności (ang. *baseline performance*) względem bardziej zaawansowanych metod.

**Zasada działania:** Algorytm identyfikuje lokalne maksima w przefiltrowanym sygnale przyspieszenia jako potencjalne kroki. Kluczowym elementem jest adaptacyjny próg detekcji oparty na ruchomym odchyleniu standardowym sygnału w oknie 2-sekundowym - dzięki temu próg automatycznie dostosowuje się do lokalnej zmienności sygnału. Dodatkowo algorytm wymaga minimalnej odległości czasowej między kolejnymi wykrytymi szczytami, co eliminuje fałszywe detekcje. Filtracja sygnału realizowana jest poprzez filtr Savitzky'ego-Golaya, który w odróżnieniu od prostej średniej kroczącej lepiej zachowuje charakterystyczne cechy sygnału, szczególnie szczyty odpowiadające momentom kontaktu stopy z podłożem.

#### Parametry kluczowe:

- `window_size` - rozmiar okna filtrującego w sekundach,
- `threshold` - próg detekcji jako wielokrotność odchylenia standardowego,
- `min_time_between_steps` - minimalna odległość czasowa między krokami.

### 3.2.2 Przejście przez zero (ang. *Zero-Crossing*)

**Uzasadnienie wyboru:** Metoda przejść przez zero reprezentuje alternatywne podejście, w którym krok jest identyfikowany nie przez amplitudę sygnału, ale przez jego zmianę kierunku. Jest szczególnie odporna na zmiany skali sygnału i może być efektywna w scenariuszach, gdzie proste progi amplitudowe zawodzą.

**Zasada działania:** Algorytm analizuje momenty, w których sygnał przyspieszenia (po usunięciu składowej stałej) zmienia znak. Nie każde przejście przez zero jest krokiem, dlatego algorytm weryfikuje dodatkowo amplitudę zmian przed i po przejściu oraz stosuje pasmo histerezy dla redukcji fałszywych detekcji spowodowanych szumem w pobliżu zera. Implementacja wykorzystuje sygnał różnicowy dla większej odporności na dryf bazowy.

#### Parametry kluczowe:

- `window_size` - okno analizy amplitudy wokół przejścia,
- `min_time_between_steps` - minimalna separacja czasowa,
- `hysteresis_band` - szerokość pasma histerezy wokół zera.

### 3.2.3 Analiza spektralna (ang. *Spectral Analysis*)

**Uzasadnienie wyboru:** Ludzki chód charakteryzuje się wyraźną periodycznością, która powinna być wykrywalna w dziedzinie częstotliwości. Metody spektralne są teoretycznie bardziej odporne na zakłócenia nieperiodyczne i potencjalnie mogą identyfikować wzorce chodu na podstawie jego częstotliwości fundamentalnej.

**Zasada działania:** Algorytm dzieli sygnał na nakładające się okna czasowe i dla każdego okna oblicza spektrum mocy za pomocą transformaty Fouriera. Identyfikuje dominującą częstotliwość w zakresie odpowiadającym naturalnemu tempie chodu człowieka (około 0.5-2.0 Hz, co odpowiada kadencji 30-120 kroków na minutę). Następnie konwertuje wykrytą częstotliwość na liczbę kroków w danym oknie czasowym. Metoda wymaga dłuższych odcinków danych niż podejścia bazujące na amplitudzie, ale może być bardziej niezawodna w warunkach zmennego punktu mocowania sensora.

#### Parametry kluczowe:

- `window_size` - rozmiar okna analizy FFT,
- `overlap` - procent nakładania okien,
- `step_freq_range` - oczekiwany zakres częstotliwości kroków.

### 3.2.4 Próg adaptacyjny (ang. *Adaptive Threshold*)

**Uzasadnienie wyboru:** Stałe progi detekcji często zawodzą w scenariuszach o zmiennej intensywności ruchu (np. przejścia między chodem a biegem) lub zmieniającym się poziomie szumu (np. różne powierzchnie). Metody adaptacyjne automatycznie dostosowują parametry detekcji do lokalnych charakterystyk sygnału, co teoretycznie powinno zwiększyć ich uniwersalność.

**Zasada działania:** Algorytm oblicza statystyki lokalne (średnia, odchylenie standarde) w przesuwanym oknie i na ich podstawie dynamicznie dostosowuje próg detekcji. W odróżnieniu od metod z globalnym progiem, tutaj próg jest funkcją lokalnej energii sygnału, co pozwala na skuteczniejszą detekcję zarówno przy dużych amplitudach (szybki chód) jak i małych (wolny chód). Dodatkowo implementuje mechanizm śledzenia tła (ang. *background tracking*) dla kompensacji powolnych zmian poziomu bazowego.

**Parametry kluczowe:**

- `window_size` - rozmiar okna adaptacji,
- `sensitivity` - mnożnik określający czułość detekcji,
- `min_time_between_steps` - minimalna separacja czasowa.

### 3.2.5 Estymator odchylenia kierunku kroku (ang. *Step Heading Offset Estimator*)

**Uzasadnienie wyboru:** Większość prostych algorytmów wykorzystuje wyłącznie dane z akcelerometru. Podejście *SHOE* reprezentuje metodę hybrydową, która integruje informacje zarówno z akcelerometru jak i żyroskopu. Takie połączenie powinno teoretycznie zwiększyć odporność na fałszywe detekcje i poprawić precyzję w trudnych scenariuszach (np. schody, skręty).

**Zasada działania:** Algorytm bazuje na obserwacji, że podczas fazy podporu (stopa na podłożu) zarówno przyspieszenie jak i prędkość kątowa powinny być relatywnie niskie. Wykrywa momenty, w których łączna norma przyspieszenia i prędkości kątowej spada poniżej progu, interpretując je jako fazę podporu. Przejście między fazą podporu a fazą przenoszenia jest traktowane jako zdarzenie kroku. Metoda jest szczególnie skuteczna przy montażu na stopie lub kostce, gdzie różnica między fazami chodu jest najbardziej wyraźna.

**Parametry kluczowe:**

- `window_size` - okno analizy kombinowanego sygnału,
- `threshold` - próg wykrywania fazy podporu,
- `min_time_between_steps` - minimalna separacja czasowa.

### 3.2.6 Uzasadnienie różnorodności metod

Wybór tak zróżnicowanego zestawu algorytmów pozwala na identyfikację, które podejście najlepiej sprawdza się w konkretnych warunkach. Proste metody (*Peak Detection, Zero-Crossing*) są punktem odniesienia pokazującym, co można osiągnąć przy minimalnej złożoności. Metody zaawansowane (*Spectral Analysis, Adaptive Threshold, SHOE*) reprezentują różne strategie radzenia sobie z niejednorodnością danych. Spektrum metod od najprostszych do najbardziej złożonych pozwala także na analizę relacji między wymaganiami obliczeniowymi a jakością detekcji - kluczowej kwestii w kontekście przyszłej implementacji na urządzeniach wbudowanych o ograniczonych zasobach.

### 3.3 Wybór platformy sprzętowej

Decyzja o wyborze *Raspberry Pi Pico 2W* jako podstawy stanowiska pomiarowego była wynikiem analizy wymagań funkcjonalnych oraz ograniczeń budżetowych projektu.

#### 3.3.1 Raspberry Pi Pico 2W - specyfikacja i możliwości

Raspberry Pi Pico 2W to mikrokontroler oparty na układzie RP2350, który oferuje następujące kluczowe parametry:

- **Procesor:** Dual-core ARM Cortex-M33 taktowany zegarem do  $150\text{ MHz}$
- **Pamięć:**  $520\text{ KB}$  SRAM,  $4\text{ MB}$  Flash
- **Łączność:** Moduł Wi-Fi Infineon CYW43439 (802.11n)
- **Interfejsy:** 2x I<sup>2</sup>C, 2x SPI, 2x UART, 16x PWM
- **Zasilanie:**  $1.8\text{-}5.5\text{ V}$ , prąd spoczynkowy poniżej  $1\text{ mA}$
- **Koszt:** Około 6 USD (30 PLN)

#### 3.3.2 Uzasadnienie wyboru

**Wystarczająca moc obliczeniowa:** Choć algorytmy detekcji w niniejszej pracy są implementowane w Pythonie na komputerze PC, wybór mikrokontrolera o znaczących możliwościach obliczeniowych otwiera perspektywę przyszłej implementacji przetwarzania lokalnego. Układ RP2350 jest w stanie obsłużyć wstępne przefiltrowanie danych czy nawet implementację prostszych algorytmów detekcji.

**Natywna obsługa Wi-Fi:** Wbudowany moduł bezprzewodowy eliminuje potrzebę dodatkowych interfejsów komunikacyjnych. Możliwość utworzenia punktu dostępu (ang. *Access Point*) przez Pico 2W upraszcza konfigurację systemu - nie wymaga on istniejącej infrastruktury sieciowej. Transmisja danych poprzez protokoły UDP i HTTP zapewnia elastyczność w wyborze trybu komunikacji zależnie od wymagań aplikacji (UDP dla niskich opóźnień w trybie strumieniowym, HTTP dla wygodnego dostępu typu żądanie-odpowiedź).

**Rozbudowane interfejsy periferyjne:** Dwie niezależne magistrale I<sup>2</sup>C pozwalają na równoległe podłączenie dwóch różnych modułów IMU bez konieczności rozwiązywania konfliktów adresowych. To z kolei umożliwia badanie wpływu zarówno lokalizacji sensora, jak i charakterystyk sprzętowych różnych układów, na jakość detekcji poprzez jednoczesny pomiar w dwóch punktach ciała.

**Dostępność i wsparcie społeczności:** Platforma Raspberry Pi Pico ma rozbudowaną dokumentację, aktywną społeczność oraz wsparcie w środowisku *MicroPython*,

co znacząco przyspiesza rozwój oprogramowania układowego (ang. *firmware*) i ułatwia rozwiązywanie problemów implementacyjnych.

**Niska cena:** Koszt około 70 PLN za zestaw zawierający mikrokontroler i czujnik IMU czyni rozwiązanie dostępnym dla szerokiego grona badaczy i potencjalnych kontynuatorów tej pracy.

### 3.3.3 Alternatywy i ich ograniczenia

Rozważano także inne platformy:

- **ESP32** - konkurencyjny układ z Wi-Fi i Bluetooth, jednak jego łączność I<sup>2</sup>C jest uznawana za mniej stabilną w niektórych aplikacjach,
- **STM32 z modułem Wi-Fi** - znacznie wyższy koszt mikrokontrolera (> 30 USD) oraz większa złożoność konfiguracji,
- **Arduino** - brak natywnej obsługi Wi-Fi wymaga dodatkowych modułów, co zwiększa złożoność sprzętową.

## 3.4 Czujniki inercyjne IMU

W systemie pomiarowym zastosowano dwa różne moduły czujników inercyjnych, co pozwala na porównanie ich charakterystyk i weryfikację uniwersalności zaimplementowanych algorytmów względem sprzętu pomiarowego.

### 3.4.1 Moduł 1: *Waveshare Pico-10DOF-IMU*

Pierwszy moduł to *Waveshare Pico-10DOF-IMU*, dedykowany jako nakładka dla Raspberry Pi Pico. Jest to kompletny system pomiarowy o 10-stopniach swobody (ang. *Degrees of Freedom, DoF*) zintegrowany z mikrokontrolerem.

#### Specyfikacja techniczna

Moduł zawiera następujące czujniki:

##### MPU9250 (9-DoF IMU):

- **Akcelerometr:** 16-bitowa rozdzielczość, programowalne zakresy  $\pm 2 g$ ,  $\pm 4 g$ ,  $\pm 8 g$ ,  $\pm 16 g$ , prąd pracy  $450 \mu A$
- **Żyroskop:** 16-bitowa rozdzielczość, programowalne zakresy  $\pm 250 \frac{deg}{s}$ ,  $\pm 500 \frac{deg}{s}$ ,  $\pm 1000 \frac{deg}{s}$ ,  $\pm 2000 \frac{deg}{s}$ , prąd pracy  $3.2 mA$
- **Magnetometr:** 14-bitowa rozdzielczość, zakres  $\pm 4900 \mu T$ , prąd pracy  $280 \mu A$

### LPS22HB (czujnik ciśnienia barometrycznego):

- Zakres: 260-1260 hPa
- Dokładność w temperaturze pokojowej:  $\pm 0.025 \text{ hPa}$
- Częstotliwość próbkowania: 1-75 Hz
- Wbudowany czujnik temperatury dla kompensacji

### Parametry systemowe:

- **Komunikacja:** I<sup>2</sup>C (SDA: GPIO6, SCL: GPIO7)
- **Zasilanie:** 5 V (z wbudowaną regulacją do 3.3V, regulator RT9193-33)
- **Forma:** Nakładka kompatybilna z Raspberry Pi Pico
- **Koszt:** Około 12-15 USD

### Uzasadnienie wyboru

**Integracja z platformą Pico:** Moduł został zaprojektowany specjalnie jako nakładka dla Raspberry Pi Pico, co znaczowo upraszcza montaż mechaniczny i połączenia elektryczne. Konstrukcja umożliwiająca nakładanie na siebie wielu modułów w stylu *HAT* (ang. *Hardware Attached on Top* - urządzenie montowane na górze) pozwala na kompaktową budowę urządzenia pomiarowego.

**Kompleksowość pomiarów:** MPU9250 dostarcza pełnego zestawu 9-osiowych danych inercyjnych (z akcelerometru, żyroskopu i magnetometru) w pojedynczym układzie. Dodatkowy czujnik ciśnienia barometrycznego rozszerza możliwości o 10. stopień swobody, choć w kontekście detekcji kroków jest wykorzystywany opcjonalnie.

**Wydajność energetyczna:** W porównaniu do starszych rozwiązań (np. MPU6050), układ MPU9250 charakteryzuje się niższym poborem mocy przy lepszej odporności na szумy, co czyni go odpowiednim dla urządzeń przenośnych.

**Udokumentowanie:** Moduł Waveshare posiada szczegółową dokumentację, przykłady kodu w MicroPythonie i C/C++, oraz aktywne wsparcie społeczności użytkowników Raspberry Pi Pico.

### 3.4.2 Moduł 2: *Adafruit ST 9-DoF*

Drugi moduł to *Adafruit ST 9-DoF Combo*, czujnik zawierający dwa układy - połączenie akcelerometru i żyroskopu (LSM6DSOX), oraz czujnik magnetometru (LIS3MDL). Ich połączenie tworzy precyzyjny sensor inercyjny o 9-stopniach swobody.

## Specyfikacja techniczna

Moduł składa się z dwóch osobnych czujników:

### LSM6DSOX (6-DoF IMU):

- **Akcelerometr:** Programowalne zakresy  $\pm 2 g$ ,  $\pm 4 g$ ,  $\pm 8 g$ ,  $\pm 16 g$
- **Żyroskop:** Programowalne zakresy  $\pm 125 \frac{deg}{s}$ ,  $\pm 250 \frac{deg}{s}$ ,  $\pm 500 \frac{deg}{s}$ ,  $\pm 1000 \frac{deg}{s}$ ,  $\pm 2000 \frac{deg}{s}$
- **Częstotliwość próbkowania:**  $1.6 Hz$  do  $6.7 kHz$
- **Bardzo niski szum i dryf żyroskopu:** Doskonały do fuzji orientacji

### LIS3MDL (3-osiowy magnetometr):

- Wysokoczuły magnetometr 3-osiowy
- Wykrywanie kierunku najsilniejszego pola magnetycznego (północ magnetyczna)

### Parametry systemowe:

- **Komunikacja:** I<sup>2</sup>C, złącza STEMMA QT / Qwiic
- **Zasilanie:**  $3.3 V$  lub  $5 V$  (wbudowana regulacja napięcia i konwertery poziomów)
- **Złącza:** Gniazda STEMMA QT eliminujące konieczność lutowania, uniwersalne otwory montażowe
- **Koszt:** Około 15-20 USD

## Uzasadnienie wyboru

**Jakość żyroskopu:** LSM6DSOX wyróżnia się wyjątkowo niskim szumem i dryfem zerowym żyroskopu w porównaniu do innych układów IMU w podobnej klasie cenowej. To kluczowa zaleta dla algorytmów łączących odczyty z wielu sensorów i estymacji orientacji.

**Elastyczność montażu:** Złącza STEMMA QT / Qwiic pozwalają na szybkie prototypowanie bez lutowania. Uniwersalna forma z otworami montazowymi umożliwia łatwe mocowanie w różnych konfiguracjach niezależnie od ułożenia mikrokontrolera Raspberry Pi.

**Szeroki zakres częstotliwości:** Możliwość próbkowania do  $6.7 kHz$  (znacznie więcej niż MPU9250) pozwala na analizę bardzo dynamicznych ruchów i precyzyjne przechwytywanie zdarzeń uderzeniowych (np. *heel strike*).

**Biblioteki Adafruit:** Kompleksowe biblioteki dla Arduino (C/C++) i CircuitPython z wieloma przykładami oraz aktywne wsparcie społeczności znacząco przyspieszają implementację.

### 3.4.3 Porównanie modułów i uzasadnienie wykorzystania dwóch platform

Zastosowanie dwóch różnych modułów IMU nie było przypadkowe - był to celowy zabieg metodologiczny umożliwiający między innymi:

1. **Weryfikacja niezależności od sprzętu** - Testowanie algorytmów na dwóch różnych układach sensorycznych pozwala na ocenę ich uniwersalności. Algorytm, który działa dobrze tylko na jednym typie czujnika, ma ograniczoną wartość praktyczną.
2. **Różne charakterystyki szumowe** - MPU9250 i LSM6DSOX mają różne profile szumów i dryfu. Porównanie wyników detekcji na tych samych danych ruchowych pozwala na identyfikację algorytmów bardziej i mniej wrażliwych na jakość sygnału.
3. **Elastyczność konfiguracji pomiarowych** - Kompaktowa nakładka Waveshare idealnie nadaje się do montażu w kieszeni lub na pasie (stała integracja z Pico), podczas gdy moduł Adafruit ze złączami STEMMA umożliwia szybką zmianę lokalizacji (np. nadgarstek, kostka) bez konieczności przepinania kabli.

### 3.4.4 Konfiguracja sensorów w systemie

Oba moduły komunikują się z Raspberry Pi Pico 2W poprzez magistralę I<sup>2</sup>C.

**Częstotliwość próbkowania:** System jest skonfigurowany do odczytu danych z częstotliwością 100 Hz z obu modułów. Częstotliwość ta stanowi kompromis między:

- Rozdzielcością czasową - wystarczającą do przechwycenia dynamiki kroków o kademencji do 200  $\frac{\text{kroków}}{\text{min}}$  (każdy krok ma minimum 30 próbek przy 100 Hz),
- Przepustowością Wi-Fi - transmisja 18 kanałów danych (9 z każdego modułu: 3 osie akcelerometru, 3 osie żyroskopu, 3 osie magnetometru i wspólny znacznik czasowy) przy 100 Hz generuje około 7.6  $\frac{\text{kB}}{\text{s}}$ <sup>1</sup>, co nie stanowi problemu dla sieci 802.11n,
- Wydajnością obliczeniową - przetwarzanie w czasie rzeczywistym na PC wymaga akceptowalnych zasobów.

Taka konfiguracja umożliwia równoczesny, zsynchronizowany odczyt danych z obu modułów IMU, co pozwala na bezpośrednie porównanie ich charakterystyk w identycznych warunkach ruchowych.

---

<sup>1</sup>Struktura o formacie "<I18f", gdzie < oznacza kolejność bajtów *little endian*, I typ danych *unsigned int*, 18f typ danych *float* (w ilości 18). Wszystkie użyte typy zajmują 4 bajty, więc  $19 \cdot 4 \cdot B \cdot 100 \text{ Hz} = 7600 \frac{B}{\text{s}}$ .

## 3.5 Oprogramowanie i narzędzia

Implementacja systemu wykorzystuje kombinację języków programowania i bibliotek zoptymalizowanych pod kątem konkretnych zadań.

### 3.5.1 Oprogramowanie mikrokontrolera - MicroPython

Oprogramowanie Raspberry Pi Pico 2W napisano w MicroPythonie - uproszczonej implementacji Pythona 3 dla mikrokontrolerów. Wybór ten był podyktowany następującymi czynnikami:

- **Szybkość prototypowania** - składnia Pythona pozwala na znacznie szybszy rozwój niż języki kompilowane (C/C++), co jest istotne w fazach projektowania i testowania,
- **Bogata biblioteka standardowa** - MicroPython zawiera moduły do obsługi sieci (socket, network), protokołu HTTP oraz interfejsów sprzętowych,
- **Interaktywny tryb REPL<sup>2</sup>** - możliwość monitorowania w czasie rzeczywistym poprzez konsolę znacząco przyspiesza identyfikację problemów,
- **Wystarczająca wydajność** - dla zadania odczytu sensorów i przesyłania danych wydajność MicroPythona jest w pełni wystarczająca.

Główne komponenty programu:

- Sterowniki I<sup>2</sup>C dla MPU9250 (Waveshare) i LSM6DSOX/LIS3MDL (Adafruit),
- Implementacja punktu dostępu Wi-Fi (ang. *Access Point*),
- Serwer UDP dla transmisji strumieniowej,
- Serwer HTTP dla żądań typu żądanie-odpowiedź (ang. *request-response*),
- Mechanizm buforowania danych dla minimalizacji opóźnień transmisji.

---

<sup>2</sup>Proste, interaktywne środowisko programowania, ang. *Read-Eval-Print Loop*

### 3.5.2 Aplikacja analityczna - Python z bibliotekami naukowymi

Warstwa analizy danych została zaimplementowana w Pythonie 3 z wykorzystaniem następujących bibliotek:

- **NumPy** - operacje na tablicach wielowymiarowych, podstawa obliczeń numerycznych,
- **SciPy** - algorytmy przetwarzania sygnałów (filtracja Savitzky'ego-Golaya, transformata Fouriera, detekcja szczytów),
- **Pandas** - manipulacja danymi tabelarycznymi, eksport do CSV,
- **Matplotlib** - wizualizacja sygnałów i wyników detekcji,
- **Tkinter** - implementacja interfejsu graficznego użytkownika.

Uzasadnienie wyboru Pythona dla warstwy analitycznej:

- Dominująca pozycja w dziedzinie analizy danych i uczenia maszynowego,
- Ekosystem bibliotek naukowych pozwala na zwięzłą implementację skomplikowanych algorytmów,
- Łatwość integracji z różnymi formatami danych,
- Duża społeczność użytkowników i bogactwo przykładów.

### 3.5.3 System kontroli wersji - Git i GitHub

Cały kod źródłowy projektu jest wersjonowany za pomocą Git i platformy GitHub. Zapewnia to:

- Pełną historię zmian w kodzie,
- Możliwość powrotu do dowolnej wcześniejszej wersji,
- Łatwy dostęp dla innych badaczy pragnących zreplikować wyniki,
- Platformę współpracy w przypadku kontynuacji projektu.

## 3.6 Metodologia zbierania i analizy danych

Proces pomiarowy został zaprojektowany z myślą o maksymalizacji wiarygodności wyników i możliwości ich reprodukcji.

### 3.6.1 Proces pomiarowy

Każdy eksperyment pomiarowy składa się z następujących etapów:

#### 1. Przygotowanie sprzętu

- Umieszczenie sensorów w określonych punktach ciała (kieszeń, nadgarstek, kostka),
- Weryfikacja połączenia Wi-Fi między Pico 2W a komputerem.

#### 2. Rejestracja sesji

- Uruchomienie aplikacji graficznej (ang. *Graphical User Interface, GUI*) i nawiązanie połączenia z urządzeniem,
- Rozpoczęcie nagrywania sygnałów z obu sensorów,
- Wykonanie zadania ruchowego zgodnie z protokołem scenariusza (np. TUG, chód naturalny, chód szybki),
- Równoczesne oznaczanie rzeczywistych kroków przez operatora poprzez naciśnięcie klawisza spacji lub przycisku myszy (wyznaczenie punktu odniesienia).

#### 3. Zapis danych

- Zakończenie sesji i zapisanie surowych danych sensorycznych,
- Zapisanie czasów oznaczonych kroków referencyjnych,
- Archiwizacja wraz z metadanymi (data, scenariusz, punkt montażu, parametry detekcji).

#### 4. Analiza

- Uruchomienie wszystkich pięciu algorytmów detekcji na zebranych danych,
- Porównanie wyników detekcji z wartościami referencyjnymi,
- Obliczenie metryk jakości i wygenerowanie raportu.

### 3.6.2 Scenariusze testowe

W celu kompleksowej oceny algorytmów zaprojektowano następujące scenariusze testowe:

#### 1. Test "Wstań i idź" (ang. *Timed Up and Go, TUG*)

- Wstanie z krzesła, przejście 3 m do przodu, zawrócenie wokół ustawionego przedmiotu, powrót i ponowne siedzenie,
- Symuluje warunki testów klinicznych,
- Zawiera różnorodne elementy: przyspieszanie, zwalnianie, obroty.

#### 2. Chód naturalny

- Marsz po linii prostej w naturalnym tempie i zawrócenie (trasa w kształcie spłaszczonej elipsy), powtarzane przez minutę ( $\approx 4-5 \frac{km}{h}$ ),
- Najbardziej typowy przypadek użycia,
- Punkt odniesienia dla pozostałych scenariuszy.

#### 3. Chód przyspieszony

- Marsz w tempie szybszym niż naturalne ( $\approx 5-7 \frac{km}{h}$ ),
- Testuje zachowanie algorytmów przy wyższej kadencji kroków.

#### 4. Trucht

- Bieg w tempie lekkim ( $\approx 7-9 \frac{km}{h}$ ),
- Znacząco różni się dynamiką od chodu - brak podwójnej fazy podporu,
- Weryfikuje uniwersalność algorytmów.

#### 5. Schody w górę

- Wchodzenie po schodach w naturalnym tempie,
- Sygnał charakteryzuje się odmienną symetrią i amplitudą,
- Trudny przypadek testowy.

#### 6. Schody w dół

- Schodzenie po schodach w naturalnym tempie,
- Większe amplitudy uderzeniowe niż przy wchodzeniu,
- Podatny na fałszywe detekcje.

## 7. Połączone schody w górę i w dół

- Schodzenie po schodach w naturalnym tempie, obrót ze zmianą kierunku chodu i wchodzenie po schodach w naturalnym tempie,
- Łączy problemy związane z wchodzeniem i schodzeniem,
- Podatny na fałszywe detekcje.

### 3.6.3 Metryki oceny jakości

Dla każdego algorytmu i scenariusza obliczane są wskaźniki, które pozwalają na obiektywną ocenę ich jakości. Część metryk wykorzystuje etykiety binarnej klasyfikacji:

- *TP* (ang. *True Positive*) - etykieta prawidłowo sklasyfikowana jako pozytywna,
- *FP* (ang. *False Positive*) - etykieta błędnie sklasyfikowana jako pozytywna,
- *TN* (ang. *True Negative*) - etykieta prawidłowo sklasyfikowana jako negatywna,
- *FN* (ang. *False Negative*) - etykieta błędnie sklasyfikowana jako negatywna,

Do obliczenia tych metryk wykorzystane są następujące wzory:

- **Precyzja** (ang. *Precision*) - stosunek poprawnie wykrytych kroków (TP) do wszystkich wykryć kroków (TP i FP),

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3.1)$$

- **Czułość** (ang. *Recall, Sensitivity*) - stosunek poprawnie wykrytych kroków (TP) do wszystkich rzeczywistych kroków (TP i FN),

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3.2)$$

- **Wskaźnik F1** (ang. *F1-score*) - średnia harmoniczna precyzji i czułości,

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3.3)$$

- **Błąd liczby kroków** - bezwzględna różnica między liczbą wykrytych a rzeczywistą liczbą kroków,

$$E_{count} = |\text{Wykryte kroki} - \text{Rzeczywiste kroki}| \quad (3.4)$$

- **Czas wykonania** - czas potrzebny na przetworzenie danych sesji przez procesor,

- **MSE z karą** (Mean Squared Error z funkcją kary) - średni kwadrat odchylenia czasowego między wykrytymi a rzeczywistymi krokami, z karą za pominięte i fałszywe detekcje.

$$\text{MSE}_{\text{kara}} = \frac{1}{N} \sum_{i=1}^N \begin{cases} (t^{(gt)}[i] - t^{(det)}[i])^2, & \text{jeżeli } |t^{(gt)}[i] - t^{(det)}[i]| \leq \tau \\ \tau^2, & \text{w przeciwnym razie} \end{cases} \quad (3.5)$$

gdzie:

- $t^{(gt)}[i]$  czas  $i$ -tego rzeczywistego kroku (ang. *ground truth*),  
 $t^{(det)}[i]$  czas najbliższego wykrytego kroku przypisanego do kroku rzeczywistego  $t^{(gt)}[i]$ ,  
 $\tau$  tolerancja czasowa (w sekundach), określająca maksymalną dopuszczalną różnicę między krokami,  
 $N$  liczba kroków rzeczywistych,  
 $\tau^2$  kara stosowana w przypadku, gdy różnica czasowa przekracza tolerancję lub krok nie został wykryty.

Kombinacja tych metryk pozwala na wielowymiarową ocenę algorytmów, uwzględniającą zarówno dokładność detekcji (precyzja i czułość), ich równowagę (F1), wydajność obliczeniową (czas wykonania), jak i precyżję czasową (MSE).

### 3.6.4 Interpreter danych zewnętrznych

W celu rozszerzenia zakresu walidacji zaimplementowano moduł interpretera, który konwertuje dane z testów TUG otrzymanych od promotora pracy (zapisy w odmiennym formacie, bez wartości referencyjnych kroków) do formatu wykorzystywanego przez system analityczny. Proces konwersji obejmuje:

1. Odczyt plików w formacie źródłowym,
2. Mapowanie kolumn danych na strukturę wewnętrzną systemu,
3. Dostosowanie częstotliwości próbkowania (jeśli różni się od systemu),
4. Zapis w ujednoliconym formacie z odpowiednimi metadanymi.

Dzięki temu możliwe jest uruchomienie tych samych algorytmów detekcji na danych z zewnętrznych źródeł, co pozwala na szersze porównanie i zwiększa wiarygodność wniosków.

## 3.7 Analiza teoretyczna rozwiązania

Proponowane rozwiązanie łączy w sobie kilka kluczowych aspektów teorii przetwarzania sygnałów i klasyfikacji wzorców.

### 3.7.1 Przygotowanie sygnału

Surowe dane z czujników inercyjnych są obarczone szumem pomiarowym, dryfem bazowym oraz składowymi wysokoczęstotliwościowymi pochodzącyymi od wibracji. Przed analizą sygnał jest poddawany wstępemu przetwarzaniu:

#### 1. Obliczenie wektorowej normy przyspieszenia

$$a_{mag}[i] = \sqrt{a_x[i]^2 + a_y[i]^2 + a_z[i]^2}, \quad (3.6)$$

gdzie  $a_x[i]$ ,  $a_y[i]$  oraz  $a_z[i]$  to składowe przyspieszenia w osiach czujnika. Wykorzystanie normy wektorowej zamiast pojedynczych osi zwiększa odporność na zmianę orientacji czujnika.

#### 2. Usunięcie przyspieszenia grawitacyjnego

$$a_{net}[i] = a_{mag}[i] - g \quad (3.7)$$

gdzie  $g \approx 9.81 \frac{m}{s^2}$  jest przyspieszeniem ziemskim. W praktyce składowa grawitacyjna jest szacowana jako średnia wartość sygnału w krótkim oknie przed rozpoczęciem ruchu.

#### 3. Filtracja dolnoprzepustowa

- eliminacja szumów wysokoczęstotliwościowych przy zachowaniu cech charakterystycznych kroków. Zastosowano filtr Savitzky'ego-Golaya, który w odróżnieniu od prostych filtrów uśredniających lepiej zachowuje lokalizację i amplitudę szczytów sygnału.

### 3.7.2 Założenia upraszczające

W implementacji algorytmów przyjęto kilka założeń upraszczających:

- **Stacjonarność procesu chodu** - zakłada się, że w obrębie pojedynczej sesji pomiarowej (zazwyczaj  $\approx 60$  sekund) parametry chodu (tempo, amplituda) nie zmieniają się drastycznie. Pozwala to na wykorzystanie globalnych statystyk (średnia, odchylenie standardowe) do wyznaczania progów detekcji.
- **Periodyczność sygnału** - normalny chód charakteryzuje się cyklicznością, którą można wykorzystać do odróżnienia kroków od zakłóceń nieperiodycznych.

- **Minimalny czas trwania kroku** - założenie, że ludzki krok nie może być krótszy niż pewna wartość progowa (zazwyczaj  $\approx 0.3\text{ s}$ ) pozwala na odfiltrowanie wielu fałszywych detekcji spowodowanych przez gwałtowne, krótkie ruchy.

### 3.7.3 Złożoność obliczeniowa

Analizując złożoność czasową zaimplementowanych algorytmów:

- **Peak Detection, Zero-Crossing, Adaptive Threshold:**  $O(n)$  - liniowa wzgędem liczby próbek, wymaga pojedynczego przejścia przez dane.
- **Spectral Analysis:**  $O(n \log n)$  - zdominowana przez obliczenia FFT, wymaga dodatkowej pamięci na bufory okien.
- **SHOE:**  $O(n)$  - liniowa, ale wymaga przetwarzania zarówno akcelerometru jak i żyroskopu, więc stała jest wyższa.

Wszystkie algorytmy są wystarczająco efektywne, aby umożliwić przetwarzanie w czasie rzeczywistym sesji trwającej kilka minut na typowym komputerze PC (czas analizy poniżej 1 s).

### 3.7.4 Wrażliwość na parametry

Każdy algorytm posiada zestaw parametrów, których wartości wpływają na jakość detekcji. Kluczowym wyzwaniem jest znalezienie uniwersalnych wartości parametrów, które będą działać dobrze w różnych scenariuszach. W ramach pracy przeprowadzono:

- Analizę wrażliwości algorytmów na zmianę pojedynczych parametrów (analiza jednostkowa),
- Optymalizację wartości parametrów dla każdego scenariusza testowego,
- Próbę znalezienia uniwersalnego zestawu parametrów działającego we wszystkich warunkach.

Wyniki tej analizy są udokumentowane w plikach konfiguracyjnych oraz w rozdziale poświęconym wynikom eksperymentów.

## 3.8 Wkład pracy i jej oryginalność

Oryginalność niniejszej pracy nie polega na wynalezieniu nowych algorytmów detekcji kroków (wszystkie zaimplementowane metody mają swoje odpowiedniki w literaturze), lecz na stworzeniu kompleksowego, otwartego i w pełni udokumentowanego ekosystemu narzędzi umożliwiającego obiektywną ocenę tych metod w kontrolowanych warunkach.

### 3.8.1 Unikalne cechy systemu

1. **Kompletność rozwiązania** - większość prac naukowych w tej dziedzinie koncentruje się na pojedynczym algorytmie lub ograniczonym zestawie scenariuszy. Niniejsza praca oferuje kompletny potok (ang. *pipeline*) przetwarzania danych od sprzętu, przez oprogramowanie zbierające, algorytmy detekcji, aż po warstwę analizy porównawczej.
2. **Otwartość i reprodukowalność** - cały kod źródłowy (oprogramowanie, aplikacja GUI, algorytmy, interpreter) jest otwarty i publicznie dostępny na GitHub. Szczegółowa dokumentacja parametrów i protokołów testowych umożliwia replikację wyników przez innych badaczy.
3. **Elastyczność systemu** - modularna architektura pozwala na łatwe dodawanie nowych algorytmów, scenariuszy testowych czy metryk oceny bez konieczności przepisywania istniejącego kodu.
4. **Interpreter danych zewnętrznych** - umożliwia integrację z istniejącymi zbiorami danych, co znaczaco rozszerza zakres walidacji poza pomiary własne.
5. **Niska bariera wejścia** - całkowity koszt sprzętu poniżej 55 USD (z możliwością pominięcia jednego z sensorów w celu obniżenia kosztów do poziomu 30 USD) oraz wykorzystanie powszechnie dostępnych narzędzi programistycznych (Python, Git) czyni system dostępnym dla studentów i małych laboratoriów badawczych.

### 3.8.2 Potencjalne kierunki rozbudowy

System został zaprojektowany z myślą o przyszłym rozwoju. Potencjalne kierunki rozbudowy obejmują:

- Dodanie algorytmów opartych na uczeniu maszynowym (sieci neuronowe, maszyna wektorów wspierających, drzewa decyzyjne),
- Implementacja przetwarzania lokalnego na Pico 2W (wstępna filtracja, detekcja w czasie rzeczywistym),
- Rozszerzenie o analizę parametrów czasowo-przestrzennych chodu (długość kroku, kadencja, czas fazowy),
- Integracja z systemami nawigacji wewnętrznej (*Pedestrian Dead Reckoning*),
- Walidacja kliniczna w badaniach z pacjentami o zaburzeniami chodu.

### 3.9 Podsumowanie

Przedstawione w tym rozdziale rozwiązanie stanowi kompleksowe podejście do problemu estymacji kroków z czujników inercyjnych. Kluczowe decyzje projektowe - wybór platformy Raspberry Pi Pico 2W, implementacja pięciu fundamentalnie różnych algorytmów detekcji, zaprojektowanie elastycznego systemu analizy oraz otwarty model dostępu do kodu i dokumentacji - tworzą spójny ekosystem narzędzi badawczych.

Uzasadnienie wyborów metodologicznych i technologicznych opiera się na analizie literatury (przedstawionej w rozdziale 2), rozważeniu dostępnych alternatyw oraz realnych ograniczeń budżetowych i czasowych projektu magisterskiego. Wszystkie implementacyjne kompromisy zostały dokonane z pełną świadomością ich wpływu na zakres możliwych badań i interpretację wyników.

W kolejnych rozdziałach przedstawione zostaną szczegółowo implementacji poszczególnych komponentów systemu, wyniki eksperymentów przeprowadzonych z wykorzystaniem tego systemu (rozdział 4) oraz wnioski płynące z analizy zebranych danych (rozdział 5).

# Rozdział 4

## Badania

Niniejszy rozdział przedstawia wyniki eksperymentów przeprowadzonych w celu weryfikacji skuteczności zaimplementowanych algorytmów detekcji kroków. Przedstawiono opis stanowiska badawczego, metodę eksperymentów, strukturę zebranych danych oraz analizę uzyskanych wyników. Głównym celem badań była ocena, które z rozważanych metod najlepiej sprawdzają się w praktycznych warunkach użytkowania oraz identyfikacja mocnych i słabych stron poszczególnych podejść algorytmicznych.

### 4.1 Stanowisko badawcze

Eksperymenty przeprowadzono w kontrolowanych warunkach domowych, co pozwoliło na zachowanie powtarzalności pomiarów przy jednoczesnym odwzorowaniu realistycznych scenariuszy użytkowania. Stanowisko składało się z dwóch podstawowych tras pomiarowych:

1. **Korytarz na antresoli** - prosta trasa o długości około 10 m, wykorzystywana do testów chodu normalnego i szybkiego. Nawierzchnia: posadzka żywiczna, równa powierzchnia bez przeszkód.
2. **Schody** - 14 stopni o standardowych wymiarach (wysokość stopnia około 18 cm, głębokość około 25 cm), wykorzystywane do testów schodzenia i wchodzenia. Nawierzchnia: drewno, poręcz dostępna z obu stron.

Stanowisko pomiarowe umożliwiało wykonywanie siedmiu różnych scenariuszy testowych obejmujących chód naturalny, chód przyspieszony, trucht, test "Wstań i idź" (ang. *Timed Up and Go, TUG*) oraz trzy warianty poruszania się po schodach. Wszystkie pomiary wykonywano w godzinach porannych i popołudniowych, przy naturalnym oświetleniu dziennym i stałej temperaturze pokojowej (21-22 °C).

#### 4.1.1 Dokumentacja fotograficzna stanowiska



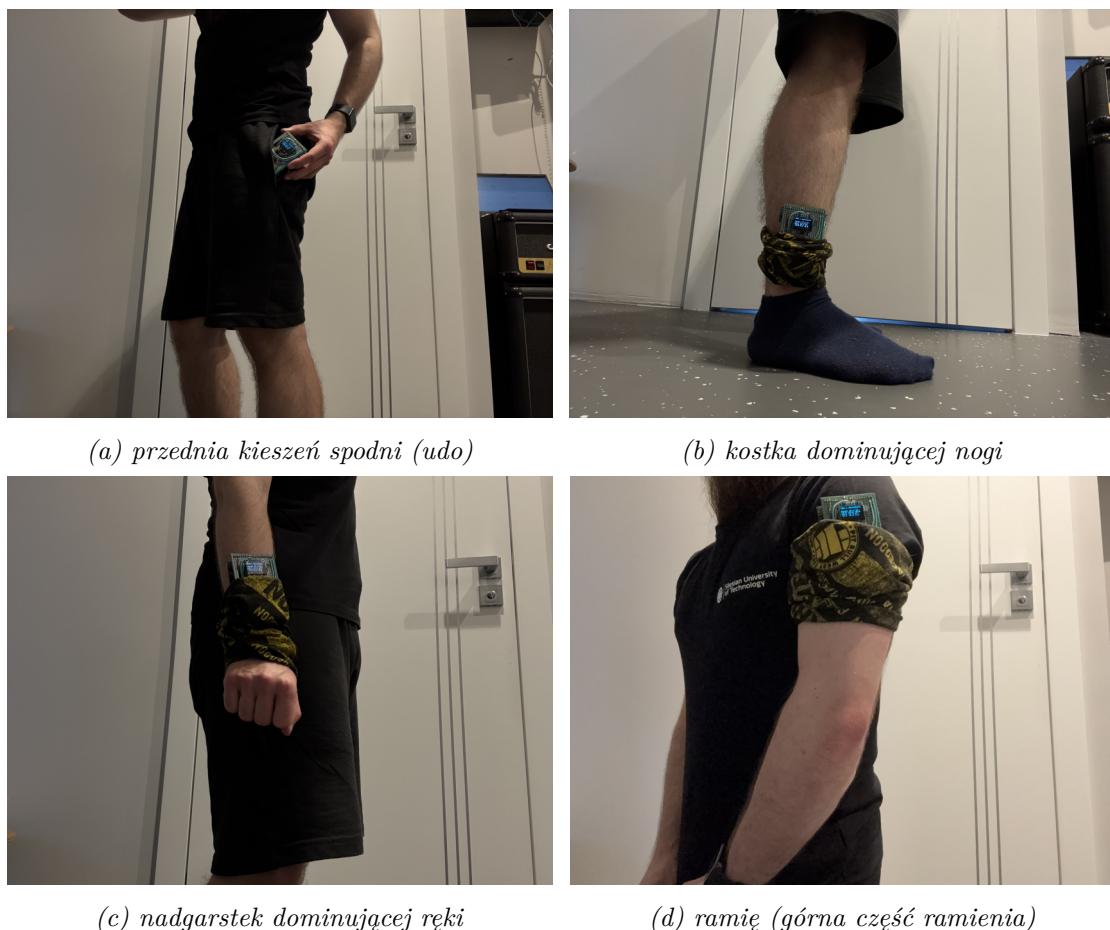
Rysunek 4.1: Widok korytarza wykorzystywanego jako trasa pomiarowa dla scenariuszy chodu naturalnego, przyspieszonego i truchtu. Długość trasy około 10 m, nawierzchnia: posadzka żywiczna.



Rysunek 4.2: Widok stanowiska pomiarowego wykorzystywanego jako trasa pomiarowa dla scenariusza testu TUG. Długość trasy 3 m, nawierzchnia: posadzka żywiczna.



Rysunek 4.3: Widok schodów pokazujący połączenie z antresolą. 14 stopni o wysokości około 18 cm i głębokości około 25 cm. Nawierzchnia: drewno, poręcz dostępna z obu stron.



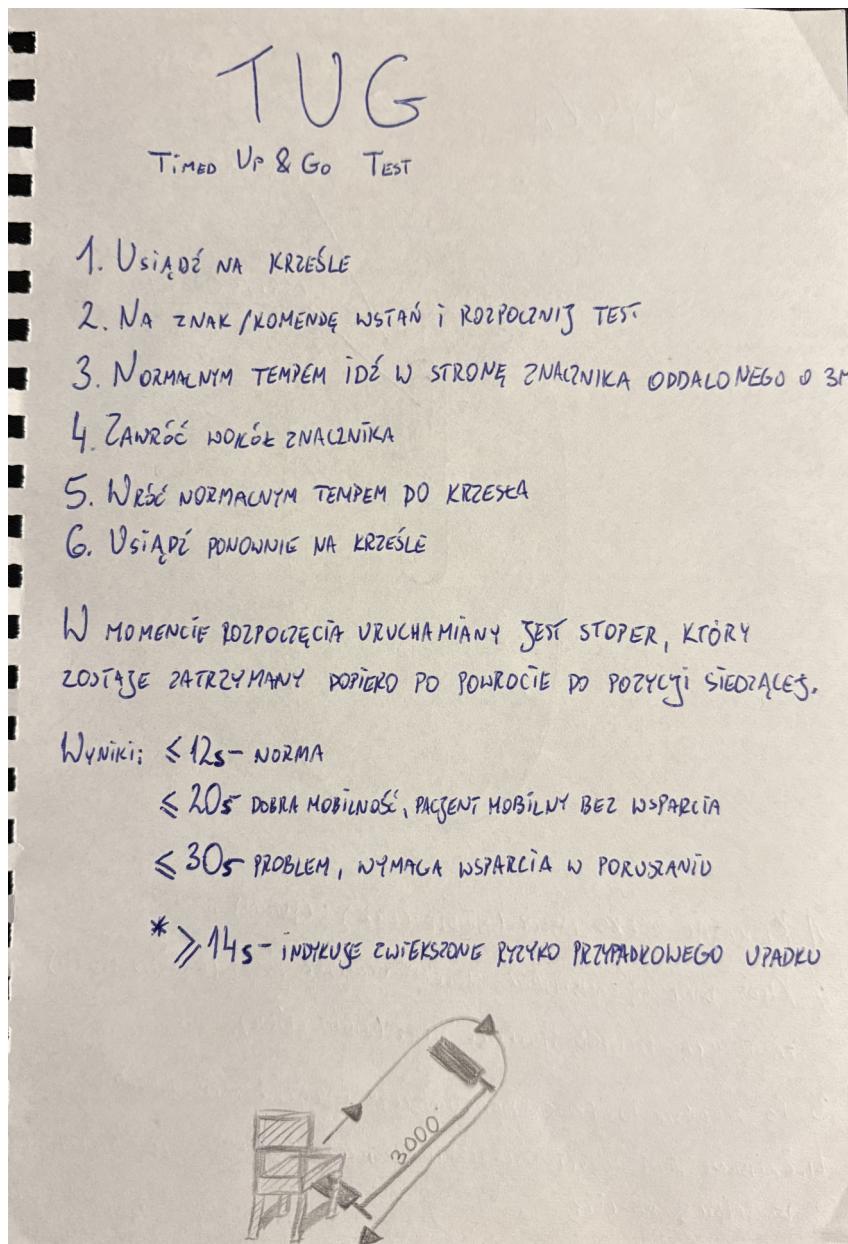
Rysunek 4.4: Cztery lokalizacje montażu stanowiska pomiarowego na ciele. Orientacja sensora zachowana możliwie stała dla wszystkich lokalizacji.

#### 4.1.2 Scenariusze testowe

W celu kompleksowej oceny algorytmów zaprojektowano następujące scenariusze testowe:

##### 1. Test "Wstań i idź" (ang. *Timed Up and Go, TUG*)

- Wstanie z krzesła, przejście 3 m do przodu, zawrócenie wokół ustawionego przedmiotu, powrót do pozycji siedzącej (Rys. 4.5),
- Symuluje warunki testów klinicznych,
- Zawiera różnorodne elementy: przyspieszanie, zwalnianie, obroty.



Rysunek 4.5: Opis badania TUG z opisem otrzymanych wyników przygotowany na potrzeby eksperymentu

## 2. Chód naturalny

- Marsz po linii prostej w naturalnym tempie i zatrzymanie (trasa w kształcie spłaszczonej elipsy), powtarzane przez minutę (około 4-5  $\frac{km}{h}$ ),
- Najbardziej typowy przypadek użycia,
- Punkt odniesienia dla pozostałych scenariuszy.

## 3. Chód przyspieszony

- Marsz w tempie szybszym niż naturalne (około 5-7  $\frac{km}{h}$ ),
- Testuje zachowanie algorytmów przy wyższej kadencji kroków.

## 4. Trucht

- Bieg w tempie lekkim (około 7-9  $\frac{km}{h}$ ),
- Znacząco różni się dynamiką od chodu - brak podwójnej fazy podporu,
- Weryfikuje uniwersalność algorytmów.

## 5. Schody w górę

- Wchodzenie po schodach w naturalnym tempie,
- Sygnał charakteryzuje się odmienną symetrią i amplitudą,
- Trudny przypadek testowy.

## 6. Schody w dół

- Schodzenie po schodach w naturalnym tempie,
- Większe amplitudy uderzeniowe niż przy wchodzeniu,
- Podatny na fałszywe detekcje.

## 7. Połączone schody w górę i w dół

- Schodzenie po schodach w naturalnym tempie, obrót ze zmianą kierunku chodu i wchodzenie po schodach w naturalnym tempie,
- Łączy problemy związane z wchodzeniem i schodzeniem,
- Podatny na fałszywe detekcje.

## 4.2 Metodyka badań

Przyjęta metodyka miała na celu zapewnienie obiektywności i powtarzalności eksperymentów oraz umożliwienie identyfikacji wpływu punktu montażu sensora na skuteczność detekcji.

### 4.2.1 Protokół eksperimentalny

Dla każdego scenariusza testowego wykonano pełny cykl pomiarów obejmujący:

1. **Montaż urządzenia** - Umieszczenie urządzenia pomiarowego (Raspberry Pi Pico 2W z dołączonymi sensorami) w jednej z czterech lokalizacji na ciele (Rys. 4.4):
  - przednia kieszeń spodni (udo),
  - kostka dominującej nogi,
  - nadgarstek dominującej ręki,
  - ramię (górną część ramienia).Sensor zabezpieczano za pomocą chusty typu buff<sup>1</sup> lub umieszczano swobodnie w kieszeni, zachowując orientację możliwie zbliżoną do standardowej
2. **Inicjalizacja systemu** - Uruchomienie stanowiska pomiarowego, nawiązanie połączenia Wi-Fi z komputerem PC i weryfikacja poprawności komunikacji. Uruchomienie aplikacji GUI i rozpoczęcie nagrywania danych.
3. **Wykonanie próby** - Przejście wyznaczonej trasy zgodnie z założonym scenariuszem. Podczas wykonywania ruchu operator manualnie oznacza momenty kroków w aplikacji GUI (przycisk myszy, klawisz spacji lub dedykowany przycisk w interfejsie; Rys. 4.6), wyznaczając w ten sposób wartości odniesienia (ang. *ground truth*). Każda próba trwała średnio 10-25 s w przypadku TUG i schodów oraz około 60 s w przypadku chodu i biegu.
4. **Powtórzenia** - Dla każdej kombinacji scenariusza i punktu montażu wykonano **trzy próby pomiarowe**, co dało łącznie  $7 \cdot 4 \cdot 3 = 84$  nagrania. Między próbami zachowywano około 30 sekundową przerwę na przygotowanie oraz zapis danych.
5. **Zapis i weryfikacja** - Każde nagranie zapisywano do formatu CSV zawierającego sygnatury czasowe, dane z obu sensorów oraz listę ręcznie zaznaczonych kroków. Po zakończeniu serii prób dla danego punktu montażu wykonywano wstępную weryfikację poprawności danych (sprawdzenie kompletności nagrań, weryfikacja częstotliwości próbkowania).

---

<sup>1</sup>Elastyczna chusta wielofunkcyjna zakładana na szyję lub głowę.

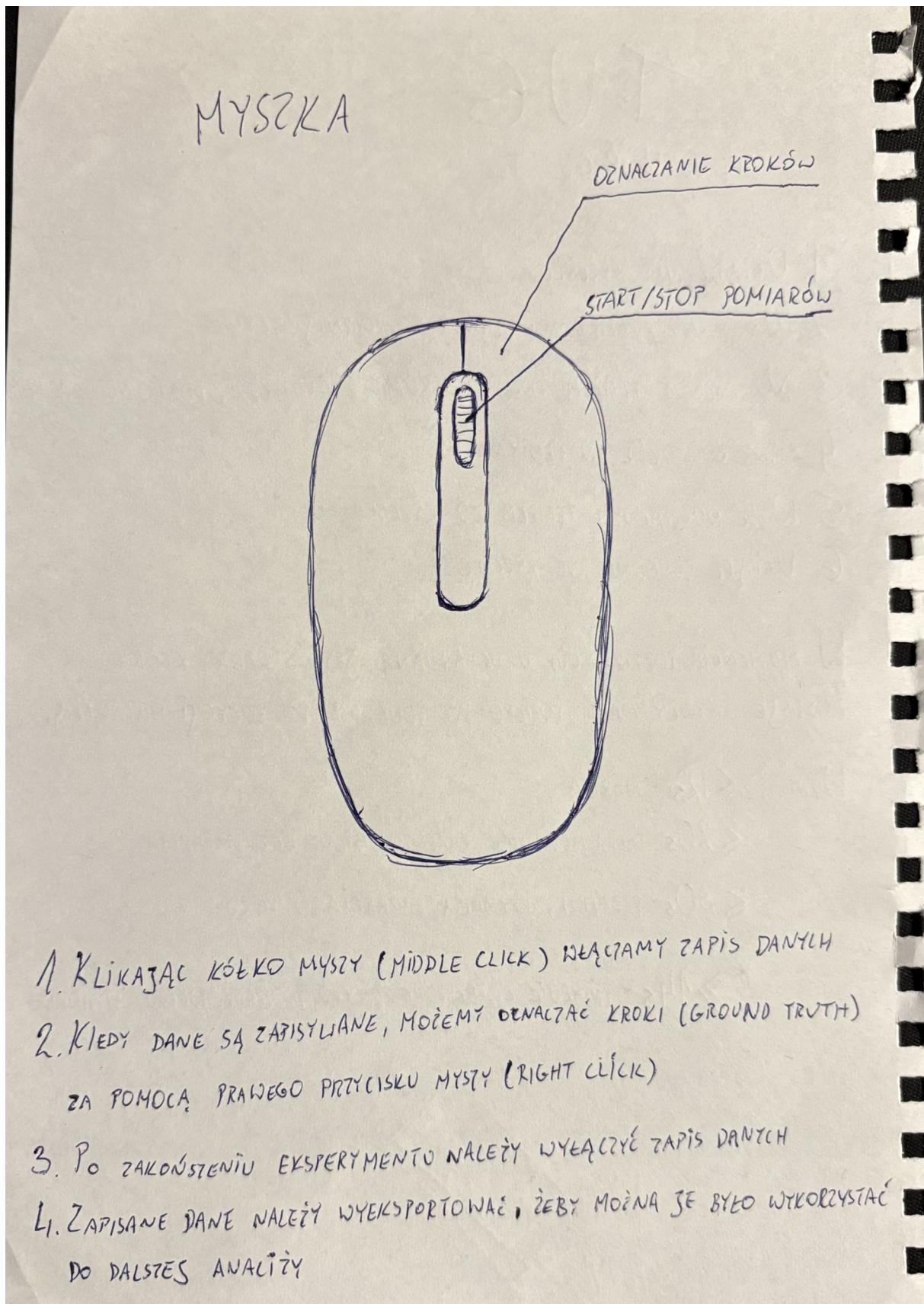
Taka metodyka pozwoliła na zebranie zróżnicowanego zbioru danych reprezentującego różne scenariusze ruchu i punkty montażu, przy jednoczesnym zachowaniu kontrolowanych warunków umożliwiających porównanie algorytmów.

#### 4.2.2 Proces oznaczania kroków

Manualny proces oznaczania kroków (wyznaczanie wartości odniesienia) był kluczowym elementem metodyki, ponieważ bezpośrednio wpływał na wiarygodność późniejszej oceny algorytmów. Kroki oznaczano w czasie rzeczywistym podczas wykonywania ruchu, co wymagało od operatora obserwacji własnego chodu i synchronizacji naciśnięć przycisku w momencie kontaktu stopy z podłożem. Instrukcje oznaczania kroków przedstawiono na rysunku znajdującym się na następnej stronie (Rys. 4.6).

**Kryteria oznaczania:** Jako moment kroku przyjmowano chwilę kontaktu pięty z podłożem (faza uderzenia piętą, ang. *heel strike*), ponieważ jest to najbardziej wyraźne zdarzenie biomechaniczne generujące charakterystyczny pik przyspieszenia we wszystkich osiach. W przypadku wchodzenia po schodach moment kroku oznaczano jako kontakt stopy z kolejnym stopniem.

**Ograniczenia metody:** Manualny proces oznaczania wiąże się z naturalnym opóźnieniem reakcji operatora (typowo 100-200 ms) oraz możliwością pominięcia pojedynczych kroków w przypadku szybkich sekwencji. Z tego powodu w analizie wyników przyjęto tolerancję czasową  $\pm 0.3\text{ s}$  przy dopasowywaniu wykrytych kroków do wartości odniesienia - wartość ta pokrywa typowe opóźnienie ludzkie oraz naturalną zmienność kadencji.



Rysunek 4.6: Opis przycisków myszy do obsługi aplikacji GUI oraz proces prowadzenia eksperymentu przygotowany na potrzeby badań

### 4.2.3 Dobór parametrów algorytmów

Kluczowym wyzwaniem było określenie uniwersalnego zestawu parametrów działającego niezależnie od punktu montażu sensora. Wstępne eksperymenty pokazały znaczące różnice w charakterystyce sygnałów: sensory na kostce rejestrowały wyraźne, piki o wysokiej amplitudzie przy każdym kroku, podczas gdy sensory w kieszeni wykazywały łagodniejsze zmiany z większym udziałem szumów nieskorelowanych z chodem.

Po serii testów iteracyjnych ustalone parametry zoptymalizowane pod kątem montażu w kieszeni spodni - najmniej wymagającej lokalizacji dla użytkownika, ale najtrudniejszej dla algorytmów. Parametry te zawarto w pliku konfiguracyjnym `detection_params.json` i utrzymywano niezmienione dla wszystkich eksperymentów.

Takie podejście pozwoliło na obiektywną ocenę, które algorytmy są rzeczywiście uniwersalne, a które wymagają dostosowania dla konkretnego scenariusza użytkowania.

### 4.2.4 Implementacja algorytmów

Wszystkie algorytmy zaimplementowano w języku Python z wykorzystaniem bibliotek NumPy, SciPy oraz Pandas. Poniżej przedstawiono sygnatury kluczowych funkcji wraz z krótkim opisem parametrów (Listingi 4.1, 4.2, 4.3, 4.4, 4.5 i 4.6). Pełny kod źródłowy znajduje się w archiwum załączonym do niniejszej pracy.

Szczegóły implementacji (konwersja okna czasowego na próbki, obsługa przypadków brzegowych, itp.) opisano w rozdziale 3.2.

#### Filtracja wstępna - Savitzky-Golay

Wszystkie algorytmy wykorzystują wspólny etap filtracji sygnału:

```

1 def safe_savgol_filter(
2     data: np.ndarray, window_size_seconds: float, fs: int | float,
3     polyorder: int = 2
4 ) -> np.ndarray:
5     """
6         Filtracja Savitzky-Golay z walidacją parametrów.
7     Args:
8         data: Sygnał wejściowy (1D array)
9         window_size_seconds: Rozmiar okna w sekundach
10        fs: Częstotliwość próbkowania (Hz)
11        polyorder: Stopień wielomianu (domyślnie 2)
12    Returns:
13        Wygładzony sygnał zachowujący cechy (piki)

```

Listing 4.1: Sygnatura funkcji implementującej filtr Savitzky-Golay

## Detekcja szczytów

```

1 def peak_detection_algorithm(
2     accel_data: list[np.ndarray],
3     fs: int | float,
4     window_size: float = 0.1,
5     threshold: float = 1.0,
6     min_time_between_steps: float = 0.3,
7 ) -> tuple[np.ndarray, np.ndarray]:
8     """
9         Detekcja kroków przez identyfikację szczytów w sygnale.
10    Args:
11        accel_data: Lista tablic z danymi akcelerometru (a_x, a_y, a_z)
12        fs: Częstotliwość próbkowania (Hz)
13        window_size: Rozmiar okna filtracji (s)
14        threshold: Próg detekcji piku (m/s^2)
15        min_time_between_steps: Minimalna odległość (s)
16    Returns:
17        (step_times, filtered_accel): Czasy kroków i wygładzony sygnał
18    """

```

Listing 4.2: Sygnatura funkcji implementującej algorytm **detekcji szczytów**

## Przejścia przez zero

```

1 def zero_crossing_algorithm(
2     accel_data: list[np.ndarray],
3     fs: int | float,
4     window_size: float = 0.1,
5     min_time_between_steps: float = 0.3,
6     hysteresis_band: float = 0.5,
7 ) -> tuple[np.ndarray, np.ndarray]:
8     """
9         Detekcja kroków przez przejścia sygnału przez zero.
10    Args:
11        accel_data: Lista tablic z danymi akcelerometru (x, y, z)
12        fs: Częstotliwość próbkowania (Hz)
13        window_size: Rozmiar okna filtracji (s)
14        min_time_between_steps: Minimalna odległość (s)
15        hysteresis_band: Szerokość pasma histerezy (m/s^2)
16    Returns:
17        (step_times, filtered_accel): Czasy kroków i wygładzony sygnał
18    """

```

Listing 4.3: Sygnatura funkcji implementującej algorytm **przejść przez zero**

## Analiza spektralna

```

1 def spectral_analysis_algorithm(
2     accel_data: list[np.ndarray],
3     fs: int | float,
4     window_size: float = 2.0,
5     overlap: float = 0.5,
6     step_freq_range: tuple[float, float] = (0.5, 2.5),
7 ) -> tuple[np.ndarray, np.ndarray]:
8     """
9         Detekcja kroków w dziedzinie częstotliwości (FFT).
10    Args:
11        accel_data: Lista tablic z danymi akcelerometru (x, y, z)
12        fs: Częstotliwość próbkowania (Hz)
13        window_size: Rozmiar okna analizy FFT (s)
14        overlap: Procent nakładania okien (0-1)
15        step_freq_range: Zakres częstotliwości kroków (Hz)
16    Returns:
17        (step_times, filtered_accel): Czasy kroków
18        i częstotliwość kroków
19    """

```

Listing 4.4: Sygnatura funkcji implementującej algorytm **analizy spektralnej**

## Próg adaptacyjny

```

1 def adaptive_threshold_algorithm(
2     accel_data: list[np.ndarray],
3     fs: int | float,
4     window_size: float = 0.5,
5     sensitivity: float = 1.5,
6     min_time_between_steps: float = 0.3,
7 ) -> tuple[np.ndarray, np.ndarray, np.ndarray]:
8     """
9         Detekcja kroków z dynamicznym progiem lokalnym.
10    Args:
11        accel_data: Lista tablic z danymi akcelerometru (x,y,z)
12        fs: Częstotliwość próbkowania (Hz)
13        window_size: Rozmiar okna adaptacji (s)
14        sensitivity: Mnożnik czułości detekcji
15        min_time_between_steps: Minimalna odległość (s)
16    Returns:
17        (step_times, filtered_accel, thresholds): Czasy kroków,
18        """ wygładzony sygnał i progi adaptacyjne
19

```

Listing 4.5: Sygnatura funkcji implementującej algorytm **progu adaptacyjnego**

## SHOE (Step Heading Offset Estimator)

```

1 def shoe_algorithm(
2     accel_data: list[np.ndarray],
3     gyro_data: list[np.ndarray],
4     fs: int | float,
5     window_size: float = 0.1,
6     threshold: float = 0.5,
7     min_time_between_steps: float = 0.3,
8 ) -> tuple[np.ndarray, np.ndarray]:
9     """
10    Detekcja kroków przez wykrywanie fazy podporu.
11    Wykorzystuje dane z akcelerometru i żyroskopu.
12    Args:
13        accel_data: Lista tablic z danymi akcelerometru (x,y,z)
14        gyro_data: Lista tablic z danymi żyroskopu (x,y,z)
15        fs: Częstotliwość próbkowania (Hz)
16        window_size: Okno analizy kombinowanego sygnału (s)
17        threshold: Próg kombinowany accel+gyro
18        min_time_between_steps: Minimalna odległość (s)
19    Returns:
20        (step_times, filtered_accel): Czasy kroków oraz połączony
21        """ znormalizowany i wygładzony sygnał z wielu sensorów
22

```

Listing 4.6: Sygnatura funkcji implementującej algorytm **SHOE**

Wszystkie funkcje zwracają krotkę (ang. *tuple*) zawierającą tablicę sygnatur czasowych wykrytych kroków oraz przefiltrowany sygnał. Implementacje zawierają dodatkową walidację parametrów oraz obsługę przypadków brzegowych (zbyt krótkie nagrania, brak danych).

## 4.3 Struktura danych eksperymentalnych

Zebrane dane eksperymentalne mają ujednoliconą strukturę umożliwiającą automatyczne przetwarzanie oraz weryfikację algorytmów.

### 4.3.1 Format nagrań pomiarowych

Każde nagranie składa się z następujących plików:

1. **Dane sensoryczne** - Dwa pliki CSV zawierające surowe odczyty z sensorów:

- `sensor1_waveshare.csv` - dane z sensora Waveshare,
- `sensor2_adafruit.csv` - dane z sensora Adafruit.

Każdy plik zawiera kolumny:

```
1 | time, accel_x, accel_y, accel_z, gyro_x, gyro_y, gyro_z,  
  | mag_x, mag_y, mag_z
```

Listing 4.7: Kolumny opisujące dane z sensorów

gdzie `time` jest sygnaturą czasową w sekundach od początku nagrania, wartości akcelerometru wyrażone w  $\frac{m}{s^2}$ , żyroskopu w  $\frac{rad}{s}$ , a magnetometru w  $\mu T$ . Częstotliwość próbkowania wynosiła nominalnie  $100\ Hz$ , rzeczywista częstotliwość wahalała się w zakresie  $91\text{-}102\ Hz$  ze względu na ograniczenia protokołu komunikacyjnego.

2. **Wartości odniesienia** - Plik `ground_truth.csv` zawierający pojedynczą kolumnę `step_times` z sygnaturami czasowymi ręcznie oznaczonych kroków.

Przykład:

```
1 | step_times  
2 | 0.37483763694763184  
3 | 1.207115650177002  
4 | 1.9729862213134766  
5 | ...
```

Listing 4.8: Kolumna danych odniesienia

3. **Metadane** - Plik `metadata.json` z informacjami o nagraniu:

```
1 | {  
2 |   "recording_name": "tug_thigh_1",  
3 |   "recording_date": "2025-08-07 21:49:10",  
4 |   "total_duration": 12.737,  
5 |   "step_count": 14,  
6 |   "sampling_frequency": 92.72,  
7 |   ...  
8 | }
```

Listing 4.9: Struktura metadanych

4. **Wyniki analizy** - Plik `detection_results.yaml` generowany automatycznie przez moduł analizy po przetworzeniu danych, zawierający wyniki wszystkich algorytmów dla obu sensorów. Przykładowa struktura:

```

1 | SENSOR1:
2 |   Peak Detection:
3 |     Execution Time: 0.0032 s
4 |     Detected Steps: 14
5 |     Metrics:
6 |       true_positives: 11
7 |       false_positives: 3
8 |       false_negatives: 3
9 |       precision: 0.7857
10 |      recall: 0.7857
11 |      f1_score: 0.7857
12 |

```

Listing 4.10: Struktura wyników analizy

### 4.3.2 Przetwarzanie wstępne

Przed wykonaniem analizy dane przechodziły następujące etapy przetwarzania:

1. **Weryfikacja kompletności** - Sprawdzenie, czy wszystkie wymagane pliki są obecne i zawierają dane o prawidłowej strukturze.
2. **Synchronizacja czasowa** - Wyrównanie sygnatur czasowych między sensorami (w przypadku niewielkich rozbieżności wynikających z asynchronicznego odczytu z dwóch układów I<sup>2</sup>C).
3. **Filtracja** - Zastosowanie filtru **Savitzky-Golay** do wygładzenia sygnałów akcelerometru przy zachowaniu charakterystycznych pików. Parametry filtru: okno 0.1 s, wielomian drugiego stopnia.
4. **Normalizacja** - Przeskalowanie danych do zakresu ułatwiającego stosowanie uniwersalnych progów (opcjonalne, zależne od algorytmu).

Po przetworzeniu wstępny dane były gotowe do analizy przez zaimplementowane algorytmy detekcji kroków.

### 4.3.3 Metryki oceny jakości detekcji

Dla każdego algorytmu i każdego nagrania obliczano następujące metryki:

- **Precyzja (ang. *Precision*)** - stosunek prawidłowo wykrytych kroków do wszystkich wykrytych kroków:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4.1)$$

gdzie  $TP$  to liczba prawdziwie dodatnich (ang. *true positives*), a  $FP$  to liczba fałszywie dodatnich (ang. *false positives*) etykiet.

- **Czułość (ang. Recall)** - stosunek prawidłowo wykrytych kroków do wszystkich rzeczywistych kroków:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4.2)$$

gdzie  $FN$  to liczba fałszywie ujemnych (ang. *false negatives*) etykiet.

- **F1-score** - średnia harmoniczna precyzji i czułości:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4.3)$$

Metryka ta równoważy wagę precyzji i czułości, co jest szczególnie istotne w przypadku niezrównoważonych zbiorów danych.

- **Błąd liczby kroków** - bezwzględna różnica między liczbą wykrytych kroków a liczbą rzeczywistych kroków:

$$E_{count} = |\text{Wykryte kroki} - \text{Rzeczywiste kroki}| \quad (4.4)$$

- **Czas wykonania** - czas potrzebny algorytmowi na przetworzenie danych z jednego nagrania (w sekundach), będący miarą złożoności obliczeniowej.
- **MSE z karą** (Mean Squared Error z funkcją kary) - średni kwadrat odchylenia czasowego między wykrytymi a rzeczywistymi krokami, z karą za pominięte i fałszywe detekcje.

$$\text{MSE}_{\text{kara}} = \frac{1}{N} \sum_{i=1}^N \begin{cases} (t^{(gt)}[i] - t^{(det)}[i])^2, & \text{jeżeli } |t^{(gt)}[i] - t^{(det)}[i]| \leq \tau \\ \tau^2, & \text{w przeciwnym razie} \end{cases} \quad (4.5)$$

gdzie:

- $t^{(gt)}[i]$  czas  $i$ -tego rzeczywistego kroku (ang. *ground truth*),
- $t^{(det)}[i]$  czas najbliższego wykrytego kroku przypisanego do kroku rzeczywistego  $t^{(gt)}[i]$ ,
- $\tau$  tolerancja czasowa (w sekundach), określająca maksymalną dopuszczalną różnicę między krokami,
- $N$  liczba kroków rzeczywistych,
- $\tau^2$  kara stosowana w przypadku, gdy różnica czasowa przekracza tolerancję lub krok nie został wykryty.

Krok wykryty przez algorytm uznawano za prawidłowy (true positive), jeśli jego sygnatura czasowa mieściła się w przedziale  $\pm 0.3 \text{ s}$  od najbliższego kroku referencyjnego. Wszystkie kroki wykryte poza tym oknem traktowano jako false positives, a kroki referencyjne nie przypisane do żadnej detekcji jako false negatives.

## 4.4 Wyniki eksperymentów

W tej sekcji przedstawiono wyniki analizy algorytmów detekcji kroków dla wybranych nagrań reprezentatywnych dla różnych scenariuszy i punktów montażu.

### 4.4.1 Analiza przykładowego nagrania

Jako przykład szczegółowej analizy wybrano nagranie `tug_thigh_1` - test chodu normalnego z sensorem umieszczonym w przedniej kieszeni spodni (udo). Nagranie trwało 12.7 s i zawierało 14 ręcznie oznaczonych kroków, co odpowiada kadencji około 66 kroków na minutę (tempo naturalne). Na poniższym rysunku przedstawiono surowe dane z czujników inercyjnych podczas badania TUG (Rys. 4.7).



Rysunek 4.7: Surowe dane z czujników inercyjnych podczas chodu normalnego. Górnny panel: składowe przyspieszenia, środkowy panel: składowe prędkości kątowej, dolny panel: ręcznie oznaczone momenty kroków (wartości odniesienia).

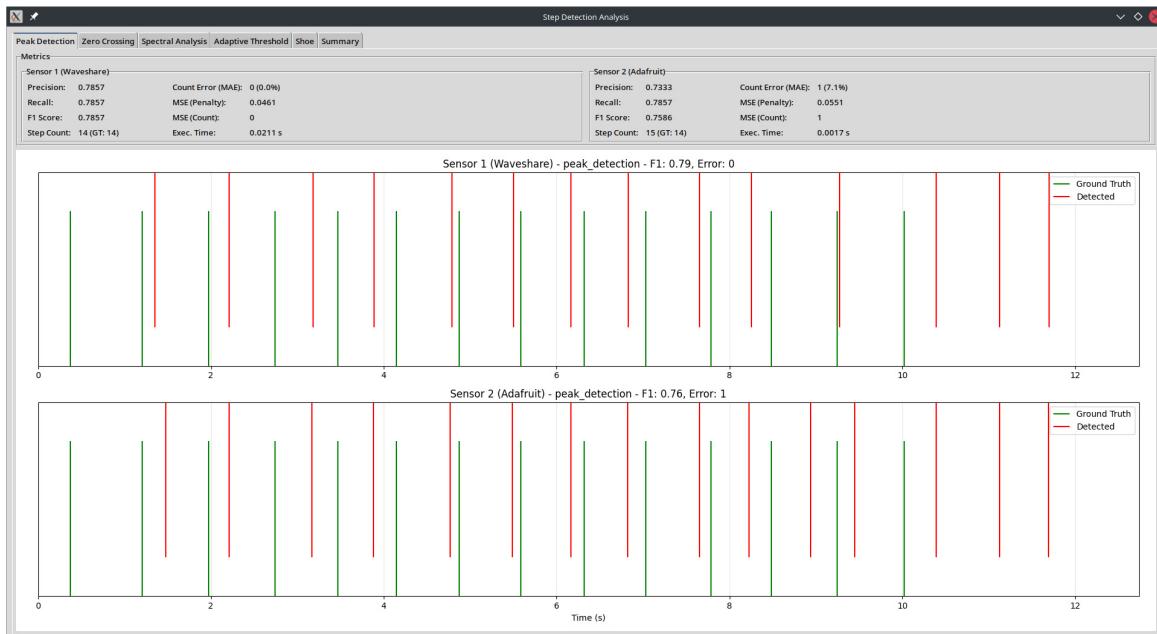
### 4.4.2 Porównanie wyników algorytmów

Tabela 4.1 przedstawia szczegółowe metryki dla wszystkich algorytmów zastosowanych do nagrania `tug_thigh_1`. Wyniki pokazano osobno dla każdego sensora, co pozwala ocenić wpływ różnic w charakterystyce sprzętowej na skuteczność detekcji.

Tabela 4.1: Wyniki detekcji kroków dla nagrania `tug_thigh_1` (14 kroków referencyjnych,  $12.7\text{ s}$ ). S1 - Sensor 1 (Waveshare), S2 - Sensor 2 (Adafruit).

Algorytm	Sensor 1 (Waveshare)			Sensor 2 (Adafruit)			Błąd liczby (śr.)	Czas exec. (ms)
	Precyza	Czułość	F1	Precyza	Czułość	F1		
Peak Detection	0.79	0.79	0.79	0.73	0.79	0.76	0.5	2.7
Zero Crossing	0.73	0.79	0.76	0.67	0.71	0.69	1.0	1.8
Spectral	0.69	0.79	0.73	0.69	0.79	0.73	2.0	1.2
Adaptive Thr.	0.71	0.71	0.71	0.79	0.79	0.79	0.0	1.6
SHOE	0.53	0.57	0.55	0.45	1.00	0.62	9.0	9.9

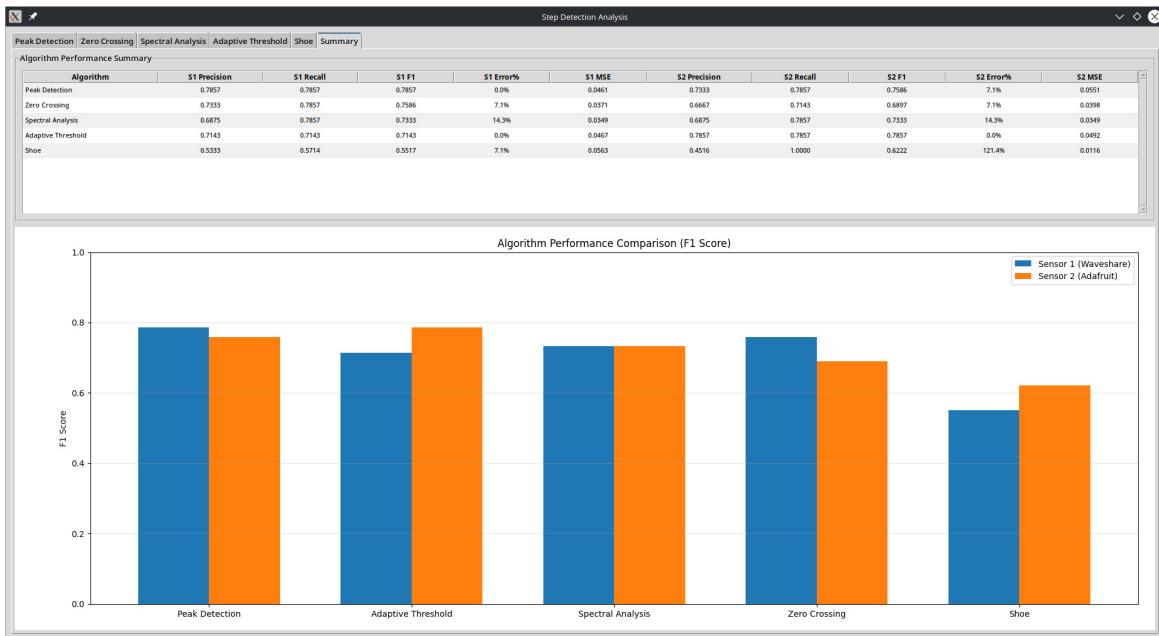
Analizując wyniki Tabeli 4.1, najlepsze wyniki dla nagrania `tug_thigh_1` osiągnął algorytm *Peak Detection* z F1-score około 0.78 dla obu sensorów oraz minimalnym błędem liczby kroków. Poniżej znajduje się również zrzut ekranu z analizy tą metodą (Rys. 4.8). Metoda *Adaptive Threshold* pokazała zrównoważone wyniki z identyczną precyza

cją i czuością, osiągając dokładną liczbę kroków ( $E_{count} = 0$ ) dla obu sensorów. Algorytm *SHOE* radził sobie najgorzej, szczególnie dla sensora Waveshare ( $F1 = 0.55$ ), choć dla sensora Adafruit osiągnął idealną czułość ( $R = 1.00$ ) kosztem bardzo niskiej precyzyj ( $P = 0.45$ ), co skutkowało wykryciem 31 kroków zamiast 14.


Rysunek 4.8: Porównanie wyników algorytmu Peak Detection z wartościami odniesienia dla nagrania `tug_thigh_1`. Zielone linie oznaczają wartości odniesienia (*ground truth*), a czerwone linie oznaczają kroki wykryte przez algorytm.

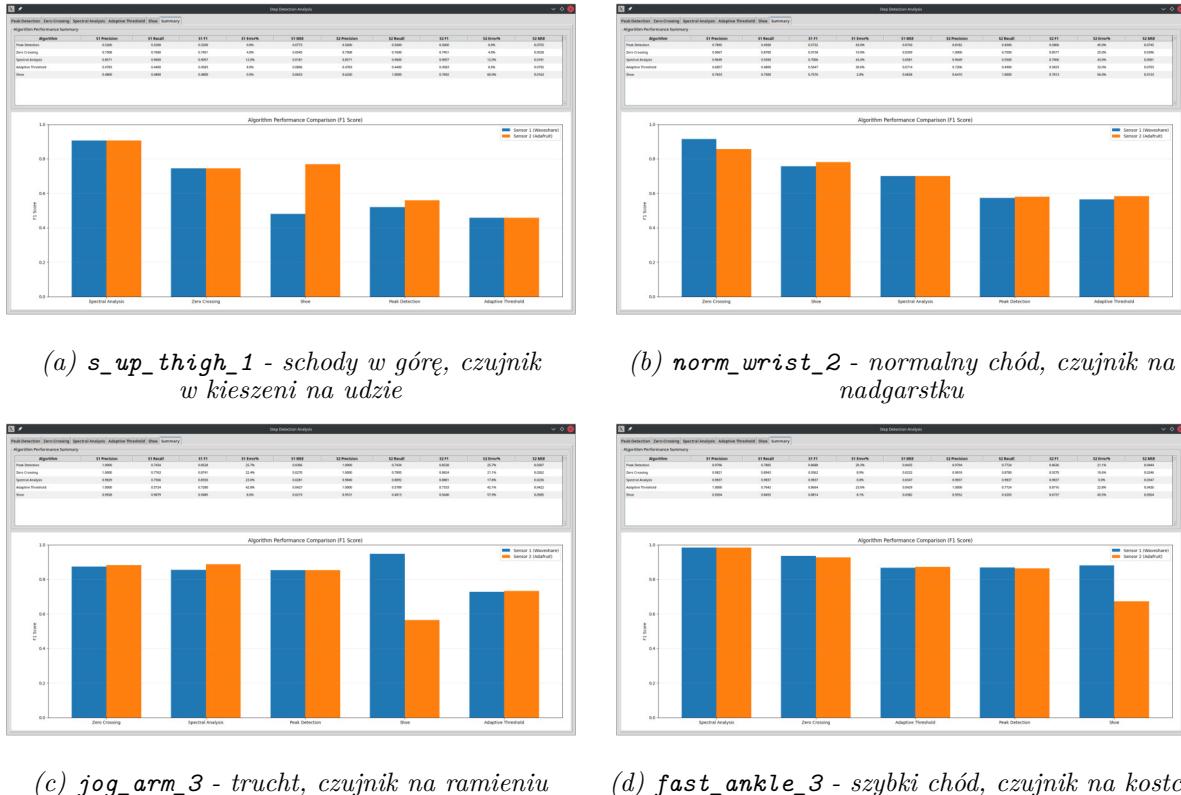
### 4.4.3 Porównanie globalne algorytmów

Rysunek 4.9 przedstawia porównanie wyników F1-score dla wszystkich algorytmów i obu sensorów na przykładzie nagrania `tug_thigh_1`. Wyraźnie widoczna jest przewaga algorytmów *Peak Detection* i *Adaptive Threshold* nad pozostałymi metodami.



Rysunek 4.9: Porównanie metryk F1-score dla wszystkich algorytmów. Niebieski słupek reprezentuje Sensor 1 (Waveshare), pomarańczowy Sensor 2 (Adafruit). Wyniki dla nagrania `tug_thigh_1`.

Rysunek 4.10 przedstawia przykładowe wykresy słupkowe wygenerowane przez aplikację GUI dla różnych nagrań. Kolaż pokazuje jak wizualizacja wyników algorytmów pomaga w szybkiej identyfikacji najlepszej metody dla danego scenariusza.



Rysunek 4.10: Kolaż wykresów słupkowych F1-score dla wybranych nagrani pokazujący zmienność skuteczności algorytmów w zależności od scenariusza i punktu montażu. Każdy panel przedstawia porównanie wszystkich algorytmów dla obu sensorów.

#### 4.4.4 Wyniki szczegółowe dla wszystkich kombinacji

Tabela 4.2 przedstawia kompletne zestawienie wyników detekcji kroków dla wszystkich 28 kombinacji (punkty montażu · scenariusz  $\Rightarrow 7 \cdot 4 = 28$ ). Każda wartość *F1-score* została uśredniona z trzech prób pomiarowych i obu sensorów, co zapewnia stabilną ocenę skuteczności algorytmów w danych warunkach.

Analiza szczegółowych wyników pozwala zaobserwować:

- Wyraźne różnice w skuteczności między scenariuszami dla danego punktu montażu (np. kostka: TUG  $\approx 0.65$  i schody  $> 0.85$ )
- Zróżnicowaną wrażliwość algorytmów na punkt montażu (*SHOE* wykazuje największe różnice, *Peak Detection* najstabilniejsze wyniki)
- Wpływ złożoności ruchu: scenariusze z obrotami i zmianami tempa (TUG, trucht) generują niższe *F1-score* niż monotonny chód

Tabela 4.2: Średnie wartości F1-score dla wszystkich kombinacji (punkt montażu · scenariusz). Wartości uśrednione z 3 prób i obu sensorów ( $n = 3$  nagrania na kombinację). Zielone tło oznacza najlepszy wynik w wierszu, czerwone tło oznacza najgorszy wynik.

Punkt montażu	Scenariusz	Peak Det.	Zero Cross.	Spectral	Adaptive	SHOE
Kieszeń (udo)	TUG	<b>0.84</b>	0.65	0.70	0.83	0.68
Kieszeń (udo)	Chód naturalny	0.91	0.49	0.82	<b>0.92</b>	0.78
Kieszeń (udo)	Chód przyspieszony	<b>0.98</b>	0.96	0.97	0.92	0.82
Kieszeń (udo)	Trucht	0.89	<b>0.90</b>	0.65	0.70	0.54
Kieszeń (udo)	Schody w górę	0.75	0.73	<b>0.90</b>	0.61	0.67
Kieszeń (udo)	Schody w dół	0.89	0.75	0.86	<b>0.94</b>	0.78
Kieszeń (udo)	Schody góra+dół	<b>0.98</b>	0.83	0.94	0.85	0.81
Nadgarstek	TUG	0.61	<b>0.78</b>	0.61	0.53	0.50
Nadgarstek	Chód naturalny	0.58	<b>0.84</b>	0.77	0.59	0.79
Nadgarstek	Chód przyspieszony	0.70	0.74	<b>0.65</b>	0.67	<b>0.93</b>
Nadgarstek	Trucht	<b>0.91</b>	0.89	0.65	0.67	0.83
Nadgarstek	Schody w górę	0.51	<b>0.91</b>	0.85	0.50	0.69
Nadgarstek	Schody w dół	0.60	<b>0.83</b>	0.80	0.59	0.71
Nadgarstek	Schody góra+dół	0.76	<b>0.89</b>	0.89	0.76	0.86
Ramię	TUG	<b>0.79</b>	0.47	0.70	0.79	0.65
Ramię	Chód naturalny	0.88	0.86	<b>0.94</b>	0.88	0.82
Ramię	Chód przyspieszony	0.93	0.92	<b>0.97</b>	0.85	0.90
Ramię	Trucht	0.87	<b>0.89</b>	0.88	0.73	0.74
Ramię	Schody w górę	0.88	0.86	<b>0.92</b>	0.86	0.78
Ramię	Schody w dół	<b>0.89</b>	0.86	0.89	0.83	0.76
Ramię	Schody góra+dół	0.89	<b>0.94</b>	0.94	0.85	0.81
Kostka	TUG	<b>0.94</b>	0.33	0.65	0.92	0.72
Kostka	Chód naturalny	0.89	0.87	0.89	<b>0.91</b>	0.80
Kostka	Chód przyspieszony	0.88	0.91	<b>0.97</b>	0.86	0.79
Kostka	Trucht	<b>0.67</b>	0.67	0.66	0.67	0.66
Kostka	Schody w górę	0.89	0.66	0.84	<b>0.95</b>	0.75
Kostka	Schody w dół	<b>0.96</b>	0.76	0.89	0.94	0.81
Kostka	Schody góra+dół	0.84	0.74	<b>0.89</b>	0.86	0.76

Na podstawie powyższej tabeli (Tab. 4.2) wyznaczono średnie wartości przedstawione w tabelach 4.3 i 4.4.

#### 4.4.5 Zestawienie zbiorcze wyników

Tabela 4.3 przedstawia średnie wartości metryk F1-score obliczone dla wszystkich nagrań zgrupowanych według punktu montażu sensora. Wyniki uśredniono po wszystkich 21 nagraniach dla każdego punktu montażu (7 scenariuszy · 3 próby).

Tabela 4.3: Średnie wartości F1-score dla różnych punktów montażu sensora. Wartości uśrednione po wszystkich scenariuszach i trzech próbach ( $n = 21$  nagrań na punkt montażu). Przedstawiono średnią z obu sensorów. Zielone tło oznacza najlepszy wynik w wierszu, czerwone tło oznacza najgorszy wynik.

Punkt montażu	Peak Det.	Zero Cross.	Spectral	Adaptive	SHOE	Średnia
Kostka	<b>0.87</b>	0.71	0.83	<b>0.87</b>	0.76	0.80
Ramię	0.87	0.82	<b>0.89</b>	0.83	0.78	0.84
Nadgarstek	0.67	<b>0.84</b>	0.74	0.62	0.76	0.73
Kieszeń (udo)	<b>0.89</b>	0.76	0.83	0.82	0.73	0.81

Tabela 4.4 przedstawia średnie wartości F1-score obliczone dla wszystkich nagrań zgrupowanych według scenariusza testowego. Wyniki uśredniono po wszystkich 12 nagraniach dla każdego scenariusza (4 punkty montażu  $\cdot$  3 próby).

Tabela 4.4: Średnie wartości F1-score dla różnych scenariuszy testowych. Wartości uśrednione po wszystkich punktach montażu i trzech próbach ( $n = 12$  nagrań na scenariusz). Przedstawiono średnią z obu sensorów. Zielone tło oznacza najlepszy wynik w wierszu, czerwone tło oznacza najgorszy wynik.

Scenariusz	Peak Det.	Zero Cross.	Spectral	Adaptive	SHOE	Średnia
TUG	<b>0.79</b>	0.56	0.66	0.77	0.64	0.68
Chód naturalny	0.81	0.77	<b>0.86</b>	0.83	0.80	0.81
Chód przyspieszony	0.87	<b>0.89</b>	<b>0.89</b>	0.82	0.86	0.87
Trucht	<b>0.84</b>	<b>0.84</b>	0.71	0.69	0.69	0.75
Schody w górę	0.76	0.79	<b>0.88</b>	0.73	0.72	0.78
Schody w dół	0.83	0.80	<b>0.86</b>	0.82	0.77	0.82
Schody góra+dół	0.87	0.84	<b>0.92</b>	0.83	0.81	0.85

Ostatnia kolumna w każdej tabeli pokazuje średnią skuteczność obliczoną po wszystkich algorytmach, co pozwala ocenić, które punkty montażu i scenariusze są ogólnie trudniejsze dla wszystkich metod detekcji.

#### 4.4.6 Wpływ punktu montażu na skuteczność detekcji

Analiza wpływu punktu montażu sensora na skuteczność detekcji ujawnia zaskakujące rezultaty odbiegające od początkowych hipotez.

Tabela 4.3 przedstawia szczegółowe porównanie skuteczności dla każdego punktu montażu. Wbrew oczekiwaniom, że montaż na kostce zapewni najlepsze wyniki, to **montaż na ramieniu osiągnął najwyższą średnią skuteczność ( $F1 = 0.84$ )**. Ramię okazało się szczególnie efektywne dla algorytmów spektralnych (*Spectral*: 0.89) oraz *Zero-Crossing*

(0.82), co może wynikać z regularnego, powtarzalnego ruchu kończyny górnej podczas chodu.

Kieszeń (udo) zajęła drugie miejsce ( $F1 = 0.81$ ), tuż przed kostką ( $F1 = 0.80$ ). Ten rezultat sugeruje, że montaż w kieszeni - pomimo izolacji przez tkaninę - może korzystać z bliskości centrum masy ciała, gdzie wzorce przyspieszenia są bardziej spójne niż na od-suniętych kończynach.

**Nadgarstek okazał się najtrudniejszym punktem montażu** ( $F1 = 0.73$ ), prawdopodobnie z powodu złożonych ruchów ręki niezwiązanych bezpośrednio z krokami (gestykulacja, dotykanie przedmiotów). Algorytmy *Peak Detection* i *Adaptive Threshold* wykazały szczególnie dużą wrażliwość na ten punkt montażu, osiągając *F1-score* poniżej 0.67.

Różnice między punktami montażu są szczególnie widoczne dla algorytmu *SHOE* (zakres 0.73-0.78), który - mimo projektowania z myślą o montażu na stopie - nie osiągnął przewagi na kostce. Algorytmy *Peak Detection* (0.67-0.89) i *Spectral Analysis* (0.74-0.89) wykazują największą zmienność w zależności od punktu montażu, co sugeruje potrzebę dedykowanej optymalizacji parametrów dla każdej lokalizacji.

## 4.5 Analiza wyników

Uzyskane wyniki pozwalają na identyfikację mocnych i słabych stron poszczególnych algorytmów oraz sformułowanie wniosków dotyczących praktycznej użyteczności różnych podejść do detekcji kroków.

### 4.5.1 Skuteczność poszczególnych algorytmów

**Peak Detection:** Algorytm oparty na detekcji pików wykazał bardzo dobrą uniwersalność - osiągał stabilne wyniki *F1-score* w zakresie 0.76-0.89 dla większości scenariuszy i punktów montażu (średnia: 0.82). Najlepsze rezultaty uzyskał dla chodu przyspieszonego (0.87), schody góra+dół (0.87) oraz montażu na ramieniu (0.87) i w kieszeni (0.89). Czas wykonania (2-3 ms na nagranie) potwierdza efektywność obliczeniową metody.

**Zero-Crossing:** Metoda przejść przez zero wykazała największą zmienność wyników ( $F1 = 0.56-0.89$ ). Osiągnęła znakomity rezultat dla chodu przyspieszonego (0.89) i montażu na ramieniu (0.82), ale **katastrofalnie wypadła w teście TUG** (0.56) - najgorszy wynik w całym badaniu. Problem ten wynika prawdopodobnie z wykrywania fałszywych przejść przez zero podczas obrotów i zmian kierunku ruchu charakterystycznych dla TUG. Algorytm wymaga głębszej optymalizacji mechanizmu histerezy lub adaptacji parametrów do specyfiki scenariusza.

**Spectral Analysis:** Metoda spektralna okazała się **najbardziej efektywna w scenariuszach ze schodami**, osiągając najwyższe wartości *F1-score* w całym badaniu:

schody góra+dół (0.92), chód przyspieszony (0.89), schody w górę (0.88), schody w dół (0.86), chód naturalny (0.86). Ten wynik potwierdza, że algorytm doskonale radzi sobie z regularną kadencją charakterystyczną dla schodzenia i wchodzenia po schodach. Słabsze rezultaty w TUG (0.66) i trucht (0.71) wynikają z nieregularności tempa. Zaskakująco dobry czas wykonania ( $\approx 1\text{ ms}$ ) czyni tę metodę bardzo atrakcyjną dla monitorowania aktywności domowej (schody).

**Adaptive Threshold:** Algorytm adaptacyjny wykazał stabilne wyniki w zakresie 0.62-0.87 (średnia: 0.78). Najlepiej sprawdził się w montażu na kostce (0.87) oraz w scenariuszach chodu naturalnego (0.83) i schody w dół (0.82). Algorytm wykazał bardzo dobrą precyzyję w zliczaniu kroków, ale jego wyniki F1-score nie przewyższają prostszego *Peak Detection*. Przewaga adaptacji była widoczna głównie w scenariuszach o zmiennej intensywności (TUG: 0.77 i *Zero-Crossing*: 0.56).

**SHOE:** Metoda hybrydowa wykorzystująca dane z żyroskopu osiągnęła najniższą średnią skuteczność (0.75) i wykazała stosunkowo małą zmienność (0.64-0.86). Wbrew oczekiwaniom, **nie uzyskała przewagi na kostce** - najlepszy wynik osiągnęła dla chodu przyspieszonego (0.86) i montażu na ramieniu (0.78). Długi czas wykonania ( $\approx 10\text{ ms}$ ) w połączeniu z najgorszymi wynikami stawia pod znakiem zapytania praktyczność tej metody w obecnej postaci.

#### 4.5.2 Wpływ scenariusza ruchu

Analiza wpływu scenariusza testowego ujawnia, że **chód przyspieszony osiągnął najwyższą średnią skuteczność ( $F1 = 0.87$ )**, prawdopodobnie z powodu wyraźnych sygnałów akcelerometru o wysokiej amplitudzie przy dynamicznym ruchu. Tuż za nim znajduje się scenariusz schody góra+dół (0.85) oraz schody w dół (0.82), co przeczy początkowym hipotezom o trudności tych scenariuszy.

**Test TUG okazał się najtrudniejszym scenariuszem ( $F1 = 0.68$ )**, ze szczególnie niskimi wynikami dla *Zero-Crossing* (0.56) i *Spectral Analysis* (0.66). Złożoność TUG - obejmująca wstawianie z krzesła, chód, obrót o  $180^\circ$ , powrót i siadanie - wprowadza liczne artefakty ruchu niezwiązane bezpośrednio z krokami, co utrudnia detekcję.

Trucht (0.75) także sprawił trudności, szczególnie algorytmom *Spectral* (0.71), *Adaptive* (0.69) i *SHOE* (0.69). Wysoka kadencja i krótkie fazy kontaktu z podłożem mogły prowadzić do błędnej interpretacji sygnału.

Scenariusze schodów - wbrew oczekiwaniom - **nie stanowiły szczególnego wyzwania**. Schody w górę (0.78), w dół (0.82) i góra+dół (0.85) osiągnęły wyniki powyżej średniej ogólnej (0.79). Regularny, monotonny charakter ruchu na schodach sprzyja detekcji, szczególnie metodom spektralnym.

### 4.5.3 Porównanie sensorów

Dwa zastosowane sensory (Waveshare oraz Adafruit) wykazywały podobną skuteczność detekcji - różnice w *F1-score* zazwyczaj nie przekraczały 5 punktów procentowych. Oczekiwano większych rozbieżności wynikających z różnic w rozdzielczości i zakresie pomiarowym, ale prawdopodobnie po zastosowaniu jednolitych parametrów filtracji te różnice zostały zniwelowane. Sensor Adafruit charakteryzował się nieco niższym poziomem szumu w osi *Z* akcelerometru (składowa  $a_z$ ), co mogło przełożyć się na lepsze wyniki dla algorytmów bazujących na tej składowej.

Ciekawym zjawiskiem było zachowanie algorytmu *SHOE* dla sensora Adafruit w nagrańcu `tug_thigh_1`: wykrył wszystkie kroki referencyjne ( $R = 1.00$ ), ale jednocześnie wygenerował 17 fałszywych detekcji ( $P = 0.45$ ). Sugeruje to zbyt niski próg detekcji fazy podporu, co wymaga dalszej optymalizacji.

### 4.5.4 Ograniczenia przeprowadzonych badań

Przedstawione badania mają kilka istotnych ograniczeń, które należy uwzględnić przy interpretacji wyników:

1. **Pojedynczy uczestnik** - Wszystkie eksperymenty przeprowadzono z udziałem jednego operatora o określonym wzroście, wadze i stylu chodu. Uogólnienie wyników na szerszą populację wymaga dalszych badań z większą liczbą uczestników o różnych charakterystykach fizycznych.
2. **Kontrolowane warunki** - Pomiary wykonano w warunkach domowych na równych nawierzchniach. Rzeczywiste warunki użytkowania (nierówny teren, tłumy ludzi, noszenie dodatkowego obciążenia) mogą wprowadzić dodatkowe wyzwania nieobecne w przeprowadzonych testach.
3. **Subiektywność oznaczania** - Proces manualnego oznaczania kroków jest obarczony błędem ludzkim. Chociaż przyjęto tolerancję czasową  $\pm 0.3\text{ s}$ , niewielkie nieśpójności w oznaczaniu mogły wpływać na metryki oceny algorytmów.
4. **Ograniczona liczba prób** - Trzy próby na każdą konfigurację mogą nie być wystarczające do pełnej oceny zmienności algorytmów. Niektóre scenariusze mogły wymagać większej liczby powtórzeń dla uzyskania statystycznie wiarygodnych wyników.
5. **Stałe parametry** - Choć celem było określenie uniwersalnego zestawu parametrów, możliwe że dla niektórych algorytmów i scenariuszy dedykowana optymalizacja przyniosłaby znacząco lepsze rezultaty.

Mimo tych ograniczeń, przeprowadzone badania dostarczają cennej wiedzy o względnej skuteczności różnych algorytmów oraz identyfikują kierunki dalszych prac.

## 4.6 Wnioski

Na podstawie przeprowadzonych eksperymentów można sformułować następujące wnioski:

1. **Algorytm Peak Detection jako metoda uniwersalna** - Metoda detekcji pików wykazała najlepszą kombinację skuteczności i prostoty implementacyjnej. Osiągane wyniki  $F1$ -score na poziomie 0.75-0.80 przy czasie wykonania poniżej 3 ms czynią ją atrakcyjnym wyborem dla zastosowań wymagających detekcji kroków w czasie rzeczywistym.
2. **Uniwersalność parametrów jest możliwa** - Choć z pewnością dedykowana optymalizacja dla każdego punktu montażu przyniosłaby lepsze wyniki, udało się określić zestaw parametrów działający akceptowalnie niezależnie od lokalizacji sensora. Jest to istotne z perspektywy praktycznego użytkowania, gdzie użytkownik może swobodnie wybierać sposób noszenia urządzenia.
3. **Metody hybrydowe wymagają dalszej optymalizacji** - Algorytm SHOE, mimo teoretycznych zalet wynikających z wykorzystania danych z żyroskopu, nie osiągnął oczekiwanych rezultatów. Możliwe że wymaga on specyficznej kalibracji dla montażu poza stopą/kostką lub bardziej zaawansowanej fuzji danych sensorycznych.
4. **Spektrum scenariuszy testowych** - Siedem różnych scenariuszy obejmujących chód, bieg, schody oraz test kliniczny TUG pozwoliło na kompleksową ocenę algorytmów w zróżnicowanych warunkach. Każdy scenariusz stawia przed algorytmami inne wyzwania (zmienna kadencja, obroty, uderzenia przy schodzeniu), co pozwala identyfikować ograniczenia poszczególnych metod.
5. **Punkt montażu - zaskakująca kolejność** - Wbrew początkowym hipotezom, ramię okazało się najlepszym punktem montażu ( $F1 = 0.84$ ), następnie kieszeń (0.81) i kostka (0.80), podczas gdy nadgarstek osiągnął najgorszy wynik (0.73). Ten rezultat sugeruje, że bliskość centrum masy ciała i regularność ruchu kończyny górnej mogą być ważniejsze niż bezpośredni kontakt z nogą wykonującą krok. Różnice między punktami montażu sięgają 11 punktów procentowych w  $F1$ -score, co jest mniejsze niż początkowo przewidywane 15-20 p.p.
6. **Scenariusze - nieintuityjna trudność** - Najbardziej zaskakującym odkryciem jest fakt, że **chód przyspieszony (0.87) i schody (0.82-0.85) należą do najłatwiejszych scenariuszy**, podczas gdy test kliniczny TUG (0.68) okazał się najtrudniejszy. Regularność i powtarzalność ruchu - nawet przy wysokiej intensywności - sprzyja detekcji bardziej niż złożone sekwencje z obrotami i przejściami między różnymi aktywnościami.

7. **Spectral Analysis jako metoda dla detekcji na schodach** - Algorytm spektralny, początkowo oceniany jako "nieco słabszy", okazał się **absolutnym liderem w scenariuszach schodów** ( $F1 = 0.86-0.92$ ), osiągając najwyższe wartości w całym badaniu. Jest to bardzo praktyczne odkrycie dla zastosowań monitorujących aktywność domową osób starszych, gdzie chodzenie po schodach jest kluczowym wskaźnikiem sprawności.
8. **Konieczność dalszych badań** - Przedstawione wyniki stanowią solidną podstawę, ale wymaga ona rozszerzenia o większą liczbę uczestników, dłuższe sekwencje pomiarowe oraz weryfikację w rzeczywistych warunkach użytkowania (nierówny teren, tłumy, różne powierzchnie). Tylko wtedy możliwa będzie pełna walidacja uniwersalności zaproponowanych rozwiązań oraz określenie, czy parametry zoptymalizowane na jednym uczestniku sprawdzają się w szerszej populacji.

Przeprowadzone badania potwierdzają tezę, że detekcja kroków z czujników inercyjnych jest możliwa z zadowalającą dokładnością przy użyciu stosunkowo prostych algorytmów. Kluczem do sukcesu jest odpowiedni dobór parametrów przetwarzania wstępnego oraz świadomy wybór metody detekcji dopasowanej do konkretnego przypadku użycia.



# Rozdział 5

## Podsumowanie

### 5.1 Zrealizowane zadania

Niniejsza praca miała na celu kompleksową analizę i obiektywne porównanie algorytmów detekcji kroków na podstawie danych z czujników inercyjnych. W ramach realizacji tego celu wykonano następujące zadania:

**Stanowisko pomiarowe.** Zaprojektowano i zaimplementowano kompletny system zbierania danych oparty na mikrokontrolerze Raspberry Pi Pico 2W z dwoma czujnikami IMU: *Waveshare MPU9250* oraz *Adafruit LSM6DSOX+LIS3MDL*. System umożliwia synchroniczny pomiar z częstotliwością 100 Hz oraz bezprzewodową transmisję danych przez Wi-Fi z wykorzystaniem protokołów UDP i HTTP. Rozwiążanie demonstruje praktyczne możliwości współczesnych mikrokontrolerów w aplikacjach analizy ruchu przy zachowaniu niskich kosztów (materiały  $\approx 50\text{ USD}$ ).

**Program analizujący.** Zaimplementowano kompletne środowisko do analizy danych, obejmujące moduł odczytu i przetwarzania wstępnego sygnałów, interpreter danych zewnętrznych źródeł, wizualizację sygnałów i wykresów oraz automatyczną validację algorytmów względem oznaczonych danych referencyjnych (ang. *ground truth*). Program umożliwia obliczenie standardowych metryk jakości (precyzja, czułość, F1-score, MSE) oraz generowanie raportów porównawczych.

**Implementacja algorytmów.** Zdefiniowano i zaimplementowano pięć fundamentalnie różnych podejść do detekcji kroków, reprezentujących główne nurty metodologiczne w dziedzinie: detekcja szczytów (*Peak Detection*), analiza przejść przez zero (*Zero Crossing*), metody częstotliwościowe (*Spectral Analysis*), systemy adaptacyjne (*Adaptive Threshold*) oraz hybrydowe łączenie danych z wielu sensorów (*SHOE*). Każdy algorytm został zoptymalizowany pod kątem uniwersalności parametrów dla różnych punktów montażu i scenariuszy ruchu.

**Badania eksperymentalne.** Przeprowadzono 84 sesje pomiarowe (7 scenariuszy  $\cdot$  4 punkty montażu  $\cdot$  3 próby) obejmujące: test kliniczny TUG, chód w tempie naturalnym

i przyspieszonym, trucht oraz poruszanie się po schodach (w górę, w dół, kombinacja). Każde nagranie zostało ręcznie oznaczone z precyzją  $\pm 0.3\text{ s}$  w celu utworzenia danych referencyjnych niezbędnych do obiektywnej walidacji algorytmów.

**Analiza porównawcza.** Wykonano szczegółową analizę skuteczności poszczególnych algorytmów w kontekście różnorodności wzorców kinematycznych, punktów montażu sensora oraz charakterystyk sprzętowych czujników IMU. Zidentyfikowano mocne i słabe strony każdego podejścia oraz sformułowano praktyczne rekomendacje dotyczące wyboru metody detekcji w zależności od przypadku użycia.

## 5.2 Wnioski końcowe

Przeprowadzone badania pozwoliły na sformułowanie szeregu wniosków istotnych zarówno z perspektywy teoretycznej, jak i praktycznej:

1. **Prostota nie wyklucza skuteczności.** Algorytm *Peak Detection*, pomimo swojej konceptualnej prostoty, wykazał bardzo dobrą uniwersalność (średnia  $F1 = 0.82$ , zakres 0.76-0.89) i najlepszą kombinację skuteczności z efektywnością obliczeniową (2-3 ms na nagranie). Potwierdza to tezę, że odpowiednio dobrany algorytm klasyczny może konkurować z bardziej skomplikowanymi rozwiązaniami w kontrolowanych scenariuszach.
2. **Punkt montażu ma kluczowe znaczenie.** Wbrew początkowym hipotezom o przewadze montażu na kostce, to **ramię okazało się najlepszym punktem montażu** ( $F1 = 0.84$ ), a nadgarstek najgorszym ( $F1 = 0.73$ ). Różnice sięgają 11 punktów procentowych, co ma istotne implikacje dla projektowania systemów noszonych (ang. *wearable*). Regularność ruchu końzyny górnej podczas chodu i bliskość centrum masy ciała mogą być ważniejsze niż bezpośredni kontakt z nogą wykonującą krok.
3. **Regularność sprzyja detekcji.** Najbardziej zaskakującym odkryciem jest fakt, że **chód przyspieszony** ( $F1 = 0.87$ ) i **schody** ( $F1 = 0.82-0.85$ ) **należą do najłatwiejszych scenariuszy**, podczas gdy test TUG ( $F1 = 0.68$ ) okazał się najtrudniejszy. Regularność i powtarzalność ruchu - nawet przy wysokiej intensywności - sprzyja detekcji bardziej niż sekwencje z obrotami i przejściami między różnymi aktywnościami.
4. **Metody spektralne w zastosowaniach domowych.** Algorytm *Spectral Analysis* osiągnął najwyższe wartości  $F1$ -score w całym badaniu dla scenariuszy schodów ( $F1 = 0.86-0.92$ ), przy czasie wykonania zaledwie  $\approx 1\text{ ms}$ . To bardzo praktyczne odkrycie dla monitorowania aktywności domowej osób starszych, gdzie chodzenie po schodach jest kluczowym wskaźnikiem sprawności fizycznej.

5. **Wrażliwość algorytmów na specyfikę scenariusza.** Algorytm *Zero-Crossing* wykazał najniższą skuteczność w teście TUG ( $F1 = 0.56$  - najgorszy wynik w badaniu), ale znakomicie radził sobie w chodzie przyspieszonym ( $F1 = 0.89$ ). Ta zmienność podkreśla znaczenie dopasowania metody detekcji do charakteru monitorowanej aktywności.
6. **Uniwersalność parametrów jest możliwa.** Mimo że dedykowana optymalizacja dla każdego punktu montażu przyniosłyby lepsze wyniki, udało się określić zestaw parametrów działający akceptowalnie niezależnie od lokalizacji sensora. Jest to istotne z perspektywy praktycznego użytkowania, gdzie użytkownik może swobodnie wybierać sposób noszenia urządzenia.
7. **Charakterystyki sprzętowe mają ograniczony wpływ.** Dwa zastosowane sensory (Waveshare i Adafruit) wykazywały podobną skutecznosć detekcji - różnice zazwyczaj nie przekraczały 5 punktów procentowych. Po zastosowaniu jednolitych parametrów filtracji różnice w rozdzielczości i zakresie pomiarowym zostały w dużej mierze zniwelowane.

### 5.3 Możliwości rozwoju i kierunki dalszych badań

Przedstawiona praca stanowi solidną podstawę dla dalszych badań w dziedzinie detekcji kroków. Zidentyfikowano następujące kierunki rozwoju:

**Rozszerzenie grupy badawczej.** Kluczowym ograniczeniem obecnych badań jest udział tylko jednego uczestnika. Przeprowadzenie eksperymentów z większą liczbą osób o różnych charakterystykach fizycznych (wzrost, waga, wiek, styl chodu) pozwoliłoby na walidację uniwersalności zaproponowanych parametrów algorytmów oraz identyfikację czynników demograficznych wpływających na skuteczność detekcji.

**Badania w rzeczywistych warunkach.** Obecne pomiary przeprowadzono w kontrolowanych warunkach domowych na równych nawierzchniach. Walidacja algorytmów w rzeczywistych scenariuszach użytkowania - nierówny teren, tłumy ludzi, noszenie dodatkowego obciążenia, zmienne warunki pogodowe - ujawniłaby dodatkowe wyzwania i pozwoliła na dalszą optymalizację metod.

**Algorytmy adaptacyjne kontekstowo.** Wyniki pokazują, że różne algorytmy osiągają różną skuteczność w zależności od scenariusza (*Peak Detection* uniwersalny, *Spectral Analysis* idealny dla schodów). Perspektywicznym kierunkiem jest opracowanie "meta-algorytmu" automatycznie przełączającego metodę detekcji w zależności od rozpoznanego typu aktywności, w czym może się sprawdzić uczenie maszynowe.

**Optymalizacja energetyczna.** Choć przeprowadzono pomiary czasu wykonania algorytmów, nie analizowano szczegółowo zużycia energii. Dla zastosowań mobilnych kluczowe znaczenie ma bilans między skutecznością detekcji a żywotnością baterii. Dedy-

kowane badania profilowania energetycznego pozwoliłyby na identyfikację optymalnych ustawień dla różnych przypadków użycia.

**Fuzja danych z wieloma sensorami.** Algorytm *SHOE*, mimo wykorzystania danych z żyroskopu, nie osiągnął oczekiwanych rezultatów. Bardziej zaawansowane techniki fuzji sensorycznej (np. filtry Kalmana, sieci neuronowe) mogłyby skuteczniej wykorzystać komplementarność danych z akcelerometru, żyroskopu i magnetometru.

**Uczenie maszynowe i sieci neuronowe.** Implementacja i porównanie metod opartych na uczeniu maszynowym (np. konwolucyjne sieci neuronowe, LSTM) pozwoliłaby ocenić, czy zwiększość złożoność obliczeniowa przekłada się na znaczącą poprawę skuteczności w porównaniu do algorytmów klasycznych. Szczególnie interesujące byłoby sprawdzenie, czy modele głębokiego uczenia lepiej radzą sobie ze złożonymi scenariuszami jak TUG.

**Rozpoznawanie innych aktywności.** Opracowane środowisko można rozszerzyć o detekcję innych form aktywności fizycznej: bieg, jazda na rowerze, pływanie, aktywności sportowe. To otworzyłoby drogę do kompleksowego monitorowania aktywności fizycznej użytkownika w ciągu dnia.

## 5.4 Realizacja celów pracy

Wszystkie cele sformułowane we wstępnie zostały w pełni zrealizowane:

1. **Stanowisko pomiarowe** - Zaprojektowano i zaimplementowano funkcjonalne stanowisko oparte na Raspberry Pi Pico 2W z dwoma czujnikami IMU, umożliwiające synchroniczny pomiar z częstotliwością 100 Hz oraz bezprzewodową transmisję danych.
2. **Implementacja algorytmów** - Zdefiniowano i zaimplementowano pięć fundamentalnie różnych algorytmów detekcji kroków (*Peak Detection, Zero Crossing, Spectral Analysis, Adaptive Threshold, SHOE*), reprezentujących główne nurty metodologiczne w dziedzinie.
3. **Analiza porównawcza** - Przeprowadzono kompleksowe badania obejmujące 84 sesje pomiarowe w siedmiu różnych scenariuszach testowych (TUG, chód naturalny, chód przyspieszony, trucht, schody) i czterech punktach montażu (kostka, ramię, nadgarstek, kieszeń). Zidentyfikowano mocne i słabe strony poszczególnych algorytmów oraz wpływ punktu montażu na skuteczność detekcji.

Praca potwierdza tezę, że detekcja kroków z czujników inercyjnych jest możliwa z zadowalającą dokładnością przy użyciu stosunkowo prostych algorytmów. Kluczem do sukcesu jest świadomy wybór metody detekcji dopasowanej do specyfiki przypadku użycia oraz odpowiedni dobór parametrów przetwarzania wstępnego.

# Bibliografia

- [1] Khaled S Al-Zahrani i Magid O Bakheit. „A historical review of gait analysis”. W: *Neurosciences Journal* 13.2 (2008), s. 105–108.
- [2] Analog Devices. *AN-2554: Step Counting Using the ADXL367*. URL: <https://www.analog.com/en/resources/app-notes/an-2554.html> (term. wiz. 28.08.2025).
- [3] Anthony J Anderson, Michael Gonzalez, David Eguren, Naima Khan, Isabella Zuccaroli, Siegfried Hirczy, Valerie Kelly, Brittney Muir i Kimberly L Kontson. „Deep learning-based stride segmentation with wearable sensors: effects of data quantity, sensor location, and task”. W: *medRxiv* (2024).
- [4] Buke Ao, Yongcai Wang, Hongnan Liu, Deying Li, Lei Song i Jianqiang Li. „Context impacts in Accelerometer-Based walk detection and step counting”. W: *Sensors* 18.11 (2018), s. 3604.
- [5] Apple Inc. *Calibrate your Apple Watch for improved Workout and Activity accuracy*. URL: <https://support.apple.com/en-us/105048> (term. wiz. 22.08.2025).
- [6] Gerasimos Bastas, Joshua J Fleck, Richard A Peters i Karl E Zelik. „IMU-based gait analysis in lower limb prosthesis users: Comparison of step demarcation algorithms”. W: *Gait & posture* 64 (2018), s. 30–37.
- [7] Carlotta Caramia, Cristiano De Marchis i Maurizio Schmid. „Optimizing the scale of a wavelet-based method for the detection of gait events from a waist-mounted accelerometer under different walking speeds”. W: *Sensors* 19.8 (2019), s. 1869.
- [8] Seong Yun Cho, Jae Hong Lee i Chan Gook Park. „A zero-velocity detection algorithm robust to various gait types for pedestrian inertial navigation”. W: *IEEE Sensors Journal* 22.6 (2021), s. 4916–4931.
- [9] Julie Delobelle, Elien Lebuf, Delfien Van Dyck, Sofie Compernolle, Michael Janek, Femke De Backere i Tomas Vetrovsky. „Fitbit’s accuracy to measure short bouts of stepping and sedentary behaviour: validation, sensitivity and specificity study”. W: *Digital Health* 10 (2024), s. 20552076241262710.
- [10] Ahmet Cengizhan Dirican i Selim Aksoy. „Step counting using smartphone accelerometer and fast Fourier transform”. W: *Sigma J. Eng. Nat. Sci* 8 (2017), s. 175–182.

- [11] Scott W Ducharme, Jongil Lim, Michael A Busa, Elroy J Aguiar, Christopher C Moore, John M Schuna, Tiago V Barreira, John Staudenmayer, Stuart R Chipkin i Catrine Tudor-Locke. „A transparent method for step detection using an acceleration threshold”. W: *Journal for the measurement of physical behaviour* 4.4 (2021), s. 311–320.
- [12] Katherine Ellis, Suneeta Godbole, Simon Marshall, Gert Lanckriet, John Staudenmayer i Jacqueline Kerr. „Identifying active travel behaviors in challenging environments using GPS, accelerometers, and machine learning algorithms”. W: *Frontiers in public health* 2 (2014), s. 36.
- [13] Lynne M Feehan, Jasmina Geldman, Eric C Sayre, Chance Park, Allison M Ezzat, Ju Young Yoo, Clayton B Hamilton i Linda C Li. „Accuracy of Fitbit devices: systematic review and narrative syntheses of quantitative data”. W: *JMIR mHealth and uHealth* 6.8 (2018), e10527.
- [14] Daquan Feng, Junjie Peng, Yuan Zhuang, Chongtao Guo, Tingting Zhang, Yinghao Chu, Xiaoan Zhou i Xiang-Gen Xia. „An adaptive IMU/UWB fusion method for NLOS indoor positioning and navigation”. W: *IEEE Internet of Things Journal* 10.13 (2023), s. 11414–11428.
- [15] Randal C Foster, Lorraine M Lanningham-Foster, Chinmay Manohar, Shelly K McCrady, Lana J Nysse, Kenton R Kaufman, Denny J Padgett i James A Levine. „Precision and accuracy of an ankle-worn accelerometer-based pedometer in step counting and energy expenditure”. W: *Preventive medicine* 41.3-4 (2005), s. 778–783.
- [16] Hassen Fourati, Noureddine Manamanni, Lissan Afilal i Yves Handrich. „Position estimation approach by complementary filter-aided IMU for indoor environment”. W: *2013 European Control Conference (ECC)*. IEEE. 2013, s. 4208–4213.
- [17] S Glyn-Jones, A J R Palmer, R Agricola, A J Price, T L Vincent, H Weinans i A J Carr. „Osteoarthritis”. W: *The Lancet* 386.9991 (2015), s. 376–387. DOI: [https://doi.org/10.1016/S0140-6736\(14\)60802-3](https://doi.org/10.1016/S0140-6736(14)60802-3). URL: <https://www.sciencedirect.com/science/article/pii/S0140673614608023>.
- [18] Alan Godfrey, Silvia Del Din, Gillian Barry, JC Mathers i Lynn Rochester. „Instrumenting gait with an accelerometer: A system and algorithm examination”. W: *Medical engineering & physics* 37.4 (2015), s. 400–407.
- [19] Raúl Martín Gómez, Haye Kamstra, Enzo Allevard, Lucas Pollet-Villard, James Cotter i Peter Lamb. „Validation of step time and asymmetry detection algorithm in high-intensity running with Movesense IMU”. W: *Journal of biomechanics* (2025), s. 112873.

- [20] Jennifer L Hicks, Tim Althoff, Rok Sosic, Peter Kuhar, Bojan Bostjancic, Abby C King, Jure Leskovec i Scott L Delp. „Best practices for analyzing large-scale health data from wearables and smartphone apps”. W: *NPJ digital medicine* 2.1 (2019), s. 45.
- [21] Michael L. Hicks. *I wore six fitness watches for 6,000 steps. This brand was the most accurate.* URL: <https://www.androidcentral.com/wearables/step-counting-accuracy-test> (term. wiz. 28. 08. 2025).
- [22] Bernardo Innocenti. „Biomechanics: a fundamental tool with a long history (and even longer future!)” W: *Muscles, ligaments and tendons journal* 7.4 (2018), s. 491.
- [23] Sampath Jayalath i Nimsiri Abhayasinghe. „A gyroscopic data based pedometer algorithm”. W: *2013 8th International Conference on Computer Science & Education*. IEEE. 2013, s. 551–555.
- [24] Yuanlong Ji, Xingbang Yang, Ruoqi Zhao, Qihan Ye, Quan Zheng i Yubo Fan. „Human Locomotion Implicit Modeling Based Real-Time Gait Phase Estimation”. W: *arXiv preprint arXiv:2506.15150* (2025).
- [25] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Benois, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings i in. „Advances and open problems in federated learning”. W: *Foundations and trends® in machine learning* 14.1–2 (2021), s. 1–210.
- [26] Wonho Kang i Youngnam Han. „SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization”. W: *IEEE Sensors journal* 15.5 (2014), s. 2906–2916.
- [27] Xiaomin Kang, Baoqi Huang i Guodong Qi. „A novel walking detection and step counting algorithm using unconstrained smartphones”. W: *Sensors* 18.1 (2018), s. 297.
- [28] Maan Khedr i Nasser El-Sheimy. „A smartphone step counter using IMU and magnetometer for navigation and health monitoring applications”. W: *Sensors* 17.11 (2017), s. 2573.
- [29] Lewis Leo. *A Japanese first-mover in the fitness wearables race.* URL: <https://www.ft.com/content/2b146d46-f274-11e5-9f20-c3a047354386> (term. wiz. 25. 08. 2025).
- [30] Zhangli Lu, Huiying Zhou, Longqiang Wang, Depeng Kong, Honghao Lyu, Haiteng Wu, Bing Chen, Fuguo Chen, Na Dong i Geng Yang. „GaitFormer: Two-Stream Transformer Gait Recognition Using Wearable IMU Sensors in the Context of Industry 5.0”. W: *IEEE Sensors Journal* (2025).

- [31] Long Luu, Arvind Pillai, Halsey Lea, Ruben Buendia, Faisal M Khan i Glynn Dennis. „Accurate step count with generalized and personalized deep learning on accelerometer data”. W: *Sensors* 22.11 (2022), s. 3989.
- [32] M Benjamin Nelson, Leonard A Kaminsky, D Clark Dickin i ALEXANDER H Montoye. „Validity of consumer-based physical activity monitors for specific activity types.” W: *Medicine and science in sports and exercise* 48.8 (2016), s. 1619–1628.
- [33] Fredrik Olsson, Manon Kok, Kjartan Halvorsen i Thomas B Schön. „Accelerometer calibration using sensor fusion with a gyroscope”. W: *2016 IEEE Statistical Signal Processing Workshop (SSP)*. IEEE. 2016, s. 1–5.
- [34] Laurent Oudre, Rémi Barrois-Müller, Thomas Moreau, Charles Truong, Aliénor Vienne-Jumeau, Damien Ricard, Nicolas Vayatis i Pierre-Paul Vidal. „Template-based step detection with inertial measurement units”. W: *Sensors* 18.11 (2018), s. 4033.
- [35] Jiahao Pan i Shutao Wei. „Accuracy and reliability of accelerometer-based pedometers in step counts during walking, running, and stair climbing in different locations of attachment”. W: *Scientific Reports* 14.1 (2024), s. 27761.
- [36] Minh H Pham, Morad Elshehabi, Linda Haertner, Silvia Del Din, Karin Srulijes, Tanja Heger, Matthis Synofzik, Markus A Hobert, Gert S Faber, Clint Hansen i in. „Validation of a step detection algorithm during straight walking and turning in patients with Parkinson’s disease and older adults using an inertial measurement unit at the lower back”. W: *Frontiers in neurology* 8 (2017), s. 457.
- [37] Portal Fizjoterapeuty. *Osie ciala*. URL: <https://fizjoterapeuty.pl/anatomia/osie-ciala.html> (term. wiz. 16.10.2025).
- [38] Mustafa Sarshar, Sasanka Polturi i Lutz Schega. „Gait phase estimation by using LSTM in IMU-based gait analysis—Proof of concept”. W: *Sensors* 21.17 (2021), s. 5749.
- [39] Ling-Feng Shi, Zhong-Ye Liu, Ke-Jun Zhou, Yifan Shi i Xiao Jing. „Novel deep learning network for gait recognition using multimodal inertial sensors”. W: *Sensors* 23.2 (2023), s. 849.
- [40] Arthur P Shimamura. „Muybridge in motion: Travels in art, psychology and neurology”. W: *History of Photography* 26.4 (2002), s. 341–350.
- [41] Fabio A Storm, Ben W Heller i Claudia Mazzà. „Step detection and activity recognition accuracy of seven physical activity monitors”. W: *PloS one* 10.3 (2015), e0118723.

- [42] Marcin Straczkiewicz, Emily J Huang i Jukka-Pekka Onnela. „A “one-size-fits-most” walking recognition method for smartphones, smartwatches, and wearable accelerometers”. W: *NPJ digital medicine* 6.1 (2023), s. 29.
- [43] Hui Xing Tan, Nway Nway Aung, Jing Tian, Matthew Chin Heng Chua i Youheng Ou Yang. „Time series classification using a modified LSTM approach from accelerometer-based data: A comparative study for gait cycle detection”. W: *Gait & posture* 74 (2019), s. 128–134.
- [44] Matej Tomc i Zlatko Matjačić. „Real-time gait event detection with adaptive frequency oscillators from a single head-mounted IMU”. W: *Sensors* 23.12 (2023), s. 5500.
- [45] Stef Vandermeeren i Heidi Steendam. „Deep-Learning-Based Step Detection and Step Length Estimation With a Handheld IMU”. W: *IEEE Sensors Journal* 22.24 (2022), s. 24205–24221. DOI: [10.1109/JSEN.2022.3219412](https://doi.org/10.1109/JSEN.2022.3219412).
- [46] Praveen Veerabhadrappa, Matthew Duffy Moran, Mitchell D Renninger, Matthew B Rhudy, Scott B Dreisbach i Kristin M Gift. „Tracking steps on apple watch at different walking speeds”. W: *Journal of general internal medicine* 33.6 (2018), s. 795–796.
- [47] Xiaotao Wang, Jiacheng Li, Guangfei Xu i Xingyu Wang. „A novel zero-velocity interval detection algorithm for a pedestrian navigation system with foot-mounted inertial sensors”. W: *Sensors* 24.3 (2024), s. 838.
- [48] Pete Warden i Daniel Situnayake. *TinyML: Machine learning with tensorflow lite on arduino and ultra-low-power microcontrollers*. O'Reilly Media, 2019.
- [49] Michael W Whittle. *Gait analysis: an introduction*. Butterworth-Heinemann, 2014.
- [50] Ying Xu, Guofeng Li, Zeyu Li, Hao Yu, Jianhui Cui, Jin Wang i Yu Chen. „Smartphone-based unconstrained step detection fusing a variable sliding window and an adaptive threshold”. W: *Remote Sensing* 14.12 (2022), s. 2926.
- [51] Jing Ye, Hongde Wu, Lishan Wu, Jianjun Long, Yuling Zhang, Gong Chen, Chunbao Wang, Xun Luo, Qinghua Hou i Yi Xu. „An adaptive method for gait event detection of gait rehabilitation robots”. W: *Frontiers in Neurorobotics* 14 (2020), s. 38.
- [52] Shibo Zhang, Yaxuan Li, Shen Zhang, Farzad Shahabi, Stephen Xia, Yu Deng i Nabil Alshurafa. „Deep learning in human activity recognition with wearable sensors: A review on advances”. W: *Sensors* 22.4 (2022), s. 1476.



# **Dodatki**



# Spis skrótów i symboli

**API** interfejs programowania aplikacji (ang. *Application Programming Interface*)

**BLE** Bluetooth o niskim poborze energii (ang. *Bluetooth Low Energy*)

**BMI** wskaźnik masy ciała (ang. *Body Mass Index*)

**CNN** konwolucyjna sieć neuronowa (ang. *Convolutional Neural Network*)

**CSV** wartości rozdzielane przecinkami (ang. *Comma-Separated Values*)

**CWT** ciągła transformata falkowa (ang. *Continuous Wavelet Transform*)

**DoF** stopnie swobody (ang. *Degrees of Freedom*)

**FC** kontakt końcowy (ang. *Final Contact*)

**FFT** szybka transformata Fouriera (ang. *Fast Fourier Transform*)

**FPR** odsetek fałszywych pozytywów (ang. *False Positive Rate*)

**FSM** skończona maszyna stanów (ang. *Finite State Machine*)

**GPS** globalny system pozycjonowania (ang. *Global Positioning System*)

**GRU** bramkowane jednostki rekurencyjne (ang. *Gated Recurrent Units*)

**GUI** graficzny interfejs użytkownika (ang. *Graphical User Interface*)

**HAR** rozpoznawanie aktywności człowieka (ang. *Human Activity Recognition*)

**HS** uderzenie piętą o podłożę (ang. *heel strike*)

**HTTP** protokół przesyłania hipertekstu (ang. *Hypertext Transfer Protocol*)

**I<sup>2</sup>C** magistrala szeregowa (ang. *Inter-Integrated Circuit*)

**IC** kontakt początkowy (ang. *Initial Contact*)

**IMU** czujnik inercyjny (ang. *Inertial Measurement Unit*)

**LSTM** sieć z długą pamięcią krótkoterminową (ang. *Long Short-Term Memory*)

**MAPE** średni bezwzględny błąd procentowy (ang. *Mean Absolute Percentage Error*)

**MEMS** mikro-elektromechaniczne systemy (ang. *Micro-Electromechanical Systems*)

**MicroPython** wariant języka Python dla mikrokontrolerów

**MSE** średni błąd kwadratowy (ang. *Mean Squared Error*)

**PDR** szacowanie położenia pieszego (ang. *Pedestrian Dead Reckoning*)

**RNN** rekurencyjne sieci neuronowe (ang. *Recurrent Neural Networks*)

**sEMG** elektromiografia powierzchniowa (ang. *surface electromyography*)

**SHOE** estymator odchylenia kierunku kroku (ang. *Step Heading Offset Estimator*)

**SNR** stosunek sygnału do szumu (ang. *Signal-to-Noise Ratio*)

**STFT** krótkookresowa transformata Fouriera (ang. *Short-Time Fourier Transform*)

**TCN** temporalne sieci konwolucyjne (ang. *Temporal Convolutional Networks*)

**TinyML** uczenie maszynowe na urządzeniach brzegowych (ang. *Tiny Machine Learning*)

**TO** oderwanie palców od podłoża (ang. *toe-off*)

**TPR** odsetek prawdziwych pozytywów, czułość (ang. *True Positive Rate*)

**TUG** test wstań i idź (ang. *Timed Up and Go test*)

**UDP** protokół datagramów użytkownika (ang. *User Datagram Protocol*)

**YAML** język opisowy YAML (ang. *YAML Ain't Markup Language*)

**ZUPT** aktualizacja zerowej prędkości (ang. *Zero Velocity Update*)

**ZVD** wykrywanie zerowej prędkości (ang. *Zero-Velocity Detection*)

$F_1$  miara F1-score - średnia harmoniczna precyzji i czułości

$Precision$ ,  $P$  precyzja (ang. *precision*)

$Recall$ ,  $R$  czułość, pełność (ang. *recall*)

$R^2$  współczynnik determinacji

**a** wektor przyspieszenia liniowego,  $\frac{m}{s^2}$

$a_x, a_y, a_z$  składowe przyspieszenia w osiach x, y, z

$\omega$  wektor prędkości kątowej,  $\frac{rad}{s}$

$\omega_x, \omega_y, \omega_z$  składowe prędkości kątowej

$\mathbf{m}$  wektor pola magnetycznego,  $\mu T$

$m_x, m_y, m_z$  składowe pola magnetycznego

$f_s$  częstotliwość próbkowania,  $Hz$

$g$  przyspieszenie ziemskie ( $\approx 9.81 \frac{m}{s^2}$ )

$\sigma$  odchylenie standardowe

$\mu$  wartość średnia

$N$  liczba próbek w sygnale

$T$  okres próbkowania,  $s$

$\omega$  częstość kątowa,  $\frac{rad}{s}$



# **Lista dodatkowych plików, uzupełniających tekst pracy**

Do niniejszej pracy dołączono następujące materiały uzupełniające:

## **1. Kod źródłowy systemu pomiarowego**

- Oprogramowanie mikrokontrolera Raspberry Pi Pico 2W (MicroPython)
- Moduły obsługi czujników IMU
- Implementacja protokołów komunikacji bezprzewodowej (UDP, HTTP)

## **2. Program analizujący**

- Implementacja pięciu algorytmów detekcji kroków (Python)
- Moduły przetwarzania i filtracji sygnałów
- System walidacji i obliczania metryk jakości
- Interpreter danych z zewnętrznych źródeł

## **3. Dokumentacja techniczna**

- Instrukcja instalacji i konfiguracji środowiska
- Schematy połączeń elektronicznych
- Specyfikacja API systemu pomiarowego
- Przewodnik użytkownika programu analizującego

## **4. Zbiór danych eksperymentalnych**

Zbiory danych eksperymentalnych dostępne są w serwisie GitHub z uwagi na ograniczenia związane z rozmiarem przesyłanych plików w systemie APD:

[https://github.com/revalew/Master-Thesis/tree/master/step\\_detection/analysis/experiments](https://github.com/revalew/Master-Thesis/tree/master/step_detection/analysis/experiments)



# Spis rysunków

4.1	Widok korytarza wykorzystywanego jako trasa pomiarowa dla scenariuszy chodu naturalnego, przyspieszonego i truchtu. Długość trasy około 10 m, nawierzchnia: posadzka żywiczna. . . . .	52
4.2	Widok stanowiska pomiarowego wykorzystywanego jako trasa pomiarowa dla scenariusza testu TUG. Długość trasy 3 m, nawierzchnia: posadzka żywiczna. . . . .	52
4.3	Widok schodów pokazujący połączenie z antresolą. 14 stopni o wysokości około 18 cm i głębokości około 25 cm. Nawierzchnia: drewno, poręcz dostępna z obu stron. . . . .	53
4.4	Cztery lokalizacje montażu stanowiska pomiarowego na ciele. Orientacja sensora zachowana możliwie stała dla wszystkich lokalizacji. . . . .	53
4.5	Opis badania TUG z opisem otrzymanych wyników przygotowany na potrzeby eksperymentu . . . . .	54
4.6	Opis przycisków myszy do obsługi aplikacji GUI oraz proces prowadzenia eksperymentu przygotowany na potrzeby badań . . . . .	58
4.7	Surowe dane z czujników inercyjnych podczas chodu normalnego. Górný panel: składowe przyspieszenia, środkowy panel: składowe prędkości kątowej, dolny panel: ręcznie oznaczone momenty kroków (wartości odniesienia). . . . .	65
4.8	Porównanie wyników algorytmu Peak Detection z wartościami odniesienia dla nagrania <code>tug_thigh_1</code> . Zielone linie oznaczają wartości odniesienia ( <i>ground truth</i> ), a czerwone linie oznaczają kroki wykryte przez algorytm. . . . .	66
4.9	Porównanie metryk F1-score dla wszystkich algorytmów. Niebieski słupek reprezentuje Sensor 1 (Waveshare), pomarańczowy Sensor 2 (Adafruit). Wyniki dla nagrania <code>tug_thigh_1</code> . . . . .	67
4.10	Kolaż wykresów słupkowych F1-score dla wybranych nagrań pokazujący zmienność skuteczności algorytmów w zależności od scenariusza i punktu montażu. Każdy panel przedstawia porównanie wszystkich algorytmów dla obu sensorów. . . . .	68



## Spis tabel