



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,
INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ

Katedra Automatyki i Inżynierii Biomedycznej

Praca dyplomowa inżynierska

Inercyjny system pomiarowy dla zdalnie sterowanego modelu

Autor:

Kierunek studiów:

Opiekun pracy:

Antoni Trąd

Automatyka i Robotyka

dr inż. Krzysztof Kolek

Kraków, 2015

Oświadczam, świadomy odpowiedzialności karnej za poświadczenie nieprawdy, że niniejszą pracę dyplomową wykonałem osobiście i samodzielnie i że nie korzystałem ze źródeł innych niż wymienione w pracy.

.....
podpis dyplomanta

Spis treści

1	Wstęp.....	5
1.1	Przedmiot problemu	5
1.2	Motywacja wyboru tematu	5
1.3	Zaprezentowane podejście rozwiązania	5
1.4	Struktura pracy	5
2	Opis tematu pracy.....	7
2.1	Istniejące rozwiązania umożliwiające lokalizowanie.....	7
2.2	Obszary zastosowań IMU.....	10
2.3	Podsumowanie rozdziału.....	11
3	Opis teoretyczny.....	13
3.1	Sposoby opisu orientacji.....	13
3.2	Filtry	16
3.3	Wprowadzone modyfikacje	19
3.4	Kalibracja.....	21
3.5	Podsumowanie rozdziału.....	23
4	Opis użytych technologii.....	25
4.1	Sprzęt.....	25
4.2	Oprogramowanie	28
5	Opis wyniku końcowego pracy	31
5.1	Elementy składowe.....	31
5.2	Fragmenty implementacji	32
5.3	Interakcje i czasowe zależności	40
5.4	Podsumowanie rozdziału.....	42
6	Testy rozwiązania.....	43
6.1	Opis doświadczeń.....	43
6.2	Przebieg i wyniki pośrednie	44
6.3	Podsumowanie końcowe	51
7	Podsumowanie	53
7.1	Problematyczne elementy.....	53
7.2	Możliwe drogi rozwoju projektu	54
8	Cytowane prace	55
9	Spis obrazków i tabel	56

1 Wstęp

1.1 Przedmiot problemu

Zagadnieniem, jakim zajęto się w tej pracy było uzyskanie informacji o położeniu, orientacji i ich zmianie w czasie. Zadanie to nie jest trywialne, ponieważ nie istnieje rozwiązanie, które byłoby jednoznacznie najlepsze. Zawsze zastosowane technologie i aparat matematyczny należy dostosować to aktualnie rozpatrywanego systemu. Lokalizowanie jest problemem z którym człowiek zmagają się od zarania dziejów. Począwszy od koczowniczych plemion gdy ważne było na przykład zapamiętanie gdzie są najżyźniejsze ziemie, poprzez różne wyprawy w nowe nieznane tereny (np. wielkie wyprawy morskie w 15 wieku). aż do dni dzisiejszych.

1.2 Motywacja wyboru tematu

Głównym powodem, dlaczego wybrano temat na pracę inżynierską, była chęć zbudowania automatycznego pilota do samolotu zdalnie sterowanego. Projekt ten jest realizowany wspólnie z Mateuszem Orłowskim i Markiem Szlachetką. Każdy z nas zajmuje się jednym z elementów potrzebnych do osiągnięcia tego celu. Takie niewielkie samoloty są narażone na stosunkowo szybkie zmiany kierunku i prędkości lotu, więc kluczowym elementem jest właśnie poprawny i odpowiednio często aktualizowany pomiar położenia. Jednocześnie należy się liczyć z ograniczonym miejscem w kadłubie i nośnością samolotu. Trzecim elementem, jaki stanowił ograniczenie był budżet na układy pomiarowe i platformę obliczeniową. Te cechy od razu spowodowały odrzucenie samego odczytu z GPS'u – wersja podstawowa była zbyt wolna i mało dokładna, natomiast RTK (opisany w rozdziale drugim) zbyt droga i duża. Początkowo sprawdzone zostało, jaką dokładność uda się uzyskać stosując same układy inercyjne. Końcowym efektem jest zastosowanie ich w połączeniu z tanim GPS'em, by uzyskać referencje do globalnego układu odniesienia i zniwelować rozbieżności powstające w czasie pracy.

1.3 Zaprezentowane podejście rozwiązania

W pracy wykorzystano układy inercyjne składające się z trójosiowych żyroskopów, magnetometrów i akcelerometrów w połączeniu z czujnikiem ciśnienia i nawigacją satelitarną. Platforma jest oparta o płytkę z czujnikami 10DOF SEN0140 firmy DFRobot [1], płytkę rozwojową STM32 Discovery z procesorem STM32M407f [2] i moduł GPS Venus638FLPx. By skutecznie połączyć czujniki wykorzystano układ filtrów (Magdwick, Mahony, Kalman, komplementarne).

1.4 Struktura pracy

W drugim rozdziale znajduje się przegląd rozwiązań istniejących na rynku, które są wykorzystywane w praktycznych sytuacjach. Porównane są ich osiągi, cechy charakterystyczne i możliwe miejsca zastosowań.

W trzecim rozdziale przedstawione zostały sposoby zapisu położenia i orientacji. Wy tłumaczono aparat matematyczny dotyczący filtrowania sygnałów i ich fuzji, w celu otrzymania stabilnych wartości. Opisane zostało też podejście, jakie wybrano do realizacji w tej pracy.

Następnie zaprezentowano platformę sprzętową i użyte czujniki. Pokazano pokrótce, jakie rodzaje sensorów są w użyciu i na czym polega zasada ich działania. Omówiono także użyte środowiska programistyczne, programy i języki.

W piątym rozdziale znajduje się opis końcowego efektu prac. Opisano płytke z mikroprocesorem, algorytmy i sposoby ich implementacji. Zamieszczono diagramy pokazujące interakcje pomiędzy elementami.

Szósty rozdział poświęcono testowaniu i sprawdzaniu działania systemu. Zawiera on opisy doświadczeń, metodologię pomiarów i analizę wyników.

W zakończeniu opisano czego nauczono się w procesie powstawania tej pracy. Wymieniono też, jakie problemy wystąpiły, oraz jakie elementy sprawiły największe trudności. Nakreślono też możliwe drogi w których ta praca będzie mogła być rozwijana.

2 Opis tematu pracy

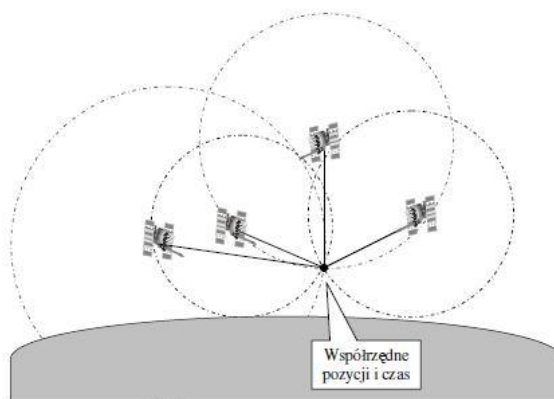
Problem lokalizacji i orientacji obiektów w przestrzeni jest często nieodłącznym tematem, przy projektowaniu mobilnego urządzenia. Wymagania jakie stawiane są systemom pomiarowym to zazwyczaj: określona dokładność, wysoka częstość odświeżania pomiaru, niski koszt czujników, energooszczędność, mały rozmiar i waga. Zazwyczaj stosuje się kilka czujników, które nawzajem kompensują swoje wady. W tym rozdziale przybliżono istniejące rozwiązania, opisując ich parametry i wady.

2.1 Istniejące rozwiązania umożliwiające lokalizowanie

- **Nawigacja satelitarna [3]:**

Pierwszym rozwiązaniem, jest GPS (Global Positioning System). Nazwa ta, mimo że w rzeczywistości odnosi się do jednego z istniejących systemów powstałych w USA („NAVSTAR GPS”), przyjęła się do ogólnego użytku. Pod tą nazwą kryją się sposoby wykorzystujące satelity w wyznaczaniu lokalizacji. Oprócz amerykańskiego równolegle działają dwa inne systemy. Rosyjski „GLONASS” i najmłodszy - europejski „Galileo”. „NAVSTAR GPS” zaczął rozwijać się dynamicznie od roku 1993. Stał się wtedy dostępny bez opłat do użytku cywilnego. Aktualnie tanie, dostępne w naszych telefonach, czy nawigacjach samochodowych GPS’y osiągają dokładność pomiarów rzędu ok. 9m (od 5m do 20m). Dużą nadzieją na poprawę rozdzielczości napawa dynamicznie rozwijający się w Europie system „Galileo” (pierwszy satelita został wystrzelony w 2011 roku). Zapowiadana dokładność jaką ten system ma udostępniać bezpłatnie to 1m.

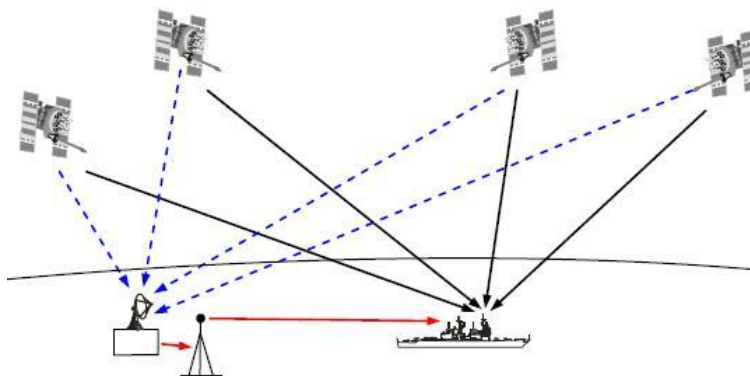
Zasada działania GPS opiera się na porównywaniu sygnałów otrzymanych z kilku satelitów. Wysyłane informacje są zsynchronizowane zegarem atomowym, który znajduje się na każdej z nich, dlatego na podstawie względnych przesunięć możemy obliczyć różnicę przebytych dróg. Dodatkowo, ponieważ satelity poruszają się po znanych orbitach możemy obliczyć ich dokładną pozycję w dowolnej chwili. Do wyznaczenia są cztery niewiadome (współrzędne i aktualny czas), dlatego konieczne jest odbieranie informacji od co najmniej czterech satelitów. Obrazowo jest to pokazane na poniższym rysunku.



Rysunek 1. Zasada działania GPS, Źródło: [4]

W temacie GPS można wyróżnić dwie podklasy systemów, które różnią się od siebie głównie dokładnością. Są to: standardowy GPS - opisany powyżej i różnicowy GPS (DGPS -

Differential GPS). Zasada działania DGPS, jest podobna do zwykłego GPS, z tym że dokładność pomiaru jest zwiększana poprzez pobieranie i wprowadzanie poprawek obliczonych na stacji referencyjnej. Stacja referencyjna to inny odbiornik GPS, który jest w znanej pozycji. Pozwala to na wyeliminowanie zewnętrznych czynników, na jakie narażony jest GPS (błąd zegara satelity, opóźnienie i zniekształcenia fali elektromagnetycznej w jonosferze i troposferze itd.). Przy odpowiedniej konfiguracji można uzyskać dokładność kilku centymetrów [5].



Rysunek 2. Zasada działania DGPS, Źródło: [4]

Ogólnymi wadami jakie występują w systemach GPS to bardzo niska częstotliwość aktualizacji pomiarów od ok. 1 Hz do 20 Hz. Dodatkowo w dynamicznych systemach, często dochodzi do zgubienia sygnału i chwilowej całkowitej straty informacji o pozycji. Jakość sygnału jest zależna od warunków atmosferycznych, czy czasu. Konieczne jest tworzenie dodatkowej infrastruktury stacji referencyjnych w wypadku DGPS. Największą wadą, która wklucza je z wielu zastosowań jest niemożliwość pracy we wnętrzach pomieszczeń.

- **Bezprzewodowa sieć czujników:**

Następnym podejściem jest wykorzystanie bezprzewodowej sieci czujników (Wireless Sensor Networks). Algorytmy z tej kategorii można podzielić na dwie grupy: odległościowe (range) i nie-odległościowe (range-free). Pierwsze sposoby polegają na obliczeniu odległości pomiędzy dwoma punktami np. odbiornik i nadajnik, a następnie wyznaczeniu bezwzględnej pozycji w sieci. Można do nich zaliczyć sposoby wykorzystujące np. moc otrzymywanego sygnału, różnice pomiędzy sygnałami odbieranym z ustalonych stacji bazowych czy czas propagacji sygnału (GPS). Metody z drugiej grupy nie obliczają dokładnej odległości do najbliższych elementów. Wykorzystują tylko informację jakie czujniki są w zasięgu. Każda z tych kategorii została dokładniej opisana w książce [6].

Jednym z przykładów czujników odległościowych jest opracowany na MIT - CLS (Cricket Location System). Działanie polega na odbieraniu informacji wysyłanej przez stacje nadawcze, których położenie dokładnie znamy. Na podstawie różnicy czasów dojścia dwóch sygnałów różnego typu (o innej długości fali) obliczana jest odległość. Rozwiązanie to oferuje dokładność rzędu 1-3 cm [7].

Problemy jakie występują w tych metodach to szumy sygnałów zakłócające precyzyjny pomiar różnicy czasu. W przestrzeniach gdzie wstępuje wiele ścian, czy przeszkód zachodzą liczne odbicia i interferencje fal. Sprawia to, że odbierane dane mogą pochodzić od dwóch

różnych stacji nadawczych, lub być znacząco zniekształcone czy opóźnione. Dodatkowo do wyznaczenia pozycji potrzeba kilku kolejnych pomiarów. Niezbędne jest zapewnienie, aby wszystkie odbyły się, gdy urządzenie jest w tym samym miejscu. Jest to niemożliwe w przypadku szybkich obiektów [8].

- **Wizyjne:**

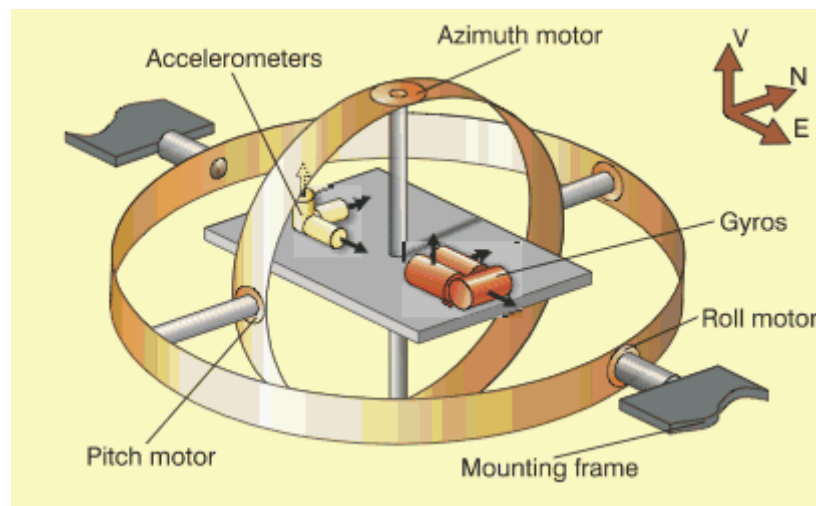
Kolejnym systemem, który wykorzystywany jest do lokalnego lokalizowania w wnętrzach pomieszczeń to układ kamer. System ten może być wykorzystywany np. do precyzyjnego lokalizowania bardzo dynamicznych obiektów jak np. quadcoptery. Taka metoda sprawdza się również w sytuacjach, gdy badane obiekty nie mogą być naruszone poprzez podpięcie dodatkowych czujników do nich, by nie zmienić ich wagi, rozmiarów, czy motoryki.

- **IMU – MEMS:**

Ostatni sposób to układy inercyjne. Często nazywane są w literaturze hasłem IMU (Inertial Measurement Units) albo MEMS (Micro Electro-Mechanical Systems). System ten stanowi fuzję co najmniej dwóch czujników: akcelerometrów mierzących przyspieszenie oraz żyroskopów mierzących prędkość kątową. Zazwyczaj jednak korzysta się też z kompasów magnetycznych mierzących pole magnetyczne ziemi. Wszystkie pomiary odbywają się w trzech osiach.

System ten w zależności od umiejscowienia czujników może pracować w dwóch konfiguracjach:

1. Stabilizowana - układ czujników nie zmienia orientacji względem układu podstawowego (ang. Stable Platform System). Jest to rozwiązanie względnie skomplikowane w budowie i wymagane jest zastosowanie serwomechanizmów (azimuth, roll, pitch motor), o bardzo dużej precyzji, które będą utrzymywały orientację czujników w układzie podstawowym. Plusem jest brak konieczności samodzielnej kompensacji grawitacji.



Rysunek 3. Konfiguracja stabilizowana IMU, Źródło: [9]

2. Niestabilizowana – orientacja układu czujników zmienia się zgodnie z orientacją układu ruchomego (ang. Strapdown System) – wymagane jest późniejsze matematyczne przekształcanie wpływu grawitacji do lokalnego układu odniesienia w celu usunięcia jego wpływu na pomiary przyspieszenia.

Położenie uzyskane tym sposobem jest tylko względne. By uzyskać wynik w współrzędnych bezwzględnych potrzeba stworzyć znany punkt odniesienia, który zarazem będzie punktem startowym (np. nasłuchując odczyt GPS). Zaletą tego rozwiązania jest duża częstotliwość aktualizacji położenia – nawet do 400Hz [10]. Główną wadą jakie posiada to rozwiązanie to zmniejszanie się dokładności w miarę upływu czasu. Skutkuje to tym, że bardzo dokładny pomiar jest możliwy tylko przez pewien, początkowy okres czasu. Wada ta jest związana między innymi z samym sposobem estymacji położenia – wykorzystujemy zawsze poprzedni pomiar, więc błędy się akumulują. Dodatkowo jest to spowodowane często błędnym ustawieniem stanu początkowego, wadami jakie posiada każdy z czujników (dryft żyroskopu i duże drgania pomiarów akcelerometru) oraz błędami obliczeniowymi (dyskretne całkowanie, błędy obciążenia itd.). Z powodu wyżej wymienionych wad zazwyczaj IMU są połączone z innym sposobem nawigacji.

Układy inercyjne można podzielić kilka podstawowych kategorii.

Tabela 1. Kategorie układów inercyjnych, Źródło: [11], [12]:

Polska nazwa:	Nawigacyjne	Taktyczne	Niskiej dokładności	Tanie (low cost MEMS)
Agielska nazwa:	Navigation grade	Tactical grade	Industrial grade	Hobbyist grade
Błąd pozycji	0,5 m/min	5,5-11 m/min	>33 m/min	>50 m/min
Cena, k€	100	20	2	<0,5
Waga	9 kg	1.5 kg	0.05 kg	<0,02kg
Przykłady	Honeywell HG9848	Honeywell HG1900	Microstrain GX2	ArduIMU

2.2 Obszary zastosowań IMU

• Geodezja:

Szerokie zastosowanie w wszystkich mobilnych systemach mapujących, fotogrametrii, skaningu laserowym (LIDAR). Pozyskanej chmurze punktów nadawana jest pozycja. W tym celu potrzeba nie tylko znać aktualne położenie systemu mapującego, ale też jego orientację. Przykładem systemu, jest np. HELIMAP [10]. Używany jest w nim GPS, układy IMU i kamera. Fuzja tych czujników pozwala uzyskać dokładność w przestrzeni rzędu 4-7cm [13].

• Medycyna – rehabilitacja:

W czasie rehabilitacji np. po udarze mózgu. W pracy [14] pokazany jest przykład wsparcia tzw. metody lustrzanej poprzez monitorowanie układem inercyjnym ruchu zdrowej ręki i kopiowanie go na niesprawną kończynę poprzez egzoszkielec. Natomiast w pracy [15] czujniki służą do obserwacji chorej części ciała i przypominaniu osobie o używaniu w przypadku długiej nieaktywności. Stanowi to motywację do długiej rehabilitacji i jest pomocą dla lekarza do lepszego doboru leczenia. W artykule [16] IMU wykorzystywane jest do łączenia sygnałów EMG (elektromiograficznych) z aktualną pozycją kończyny. Wyniki mogą służyć do tworzenia bionicznych rąk, które były by sterowane bezpośrednio z zakończeń układu nerwowego.

• Systemy nawigacji:

Wspomaganie pracy GPSu w momentach, gdy ten zagubił sygnał, wykrywanie błędnego odczytu przy pracy z niskiej jakości nawigacją [17]. Bardzo popularnym rozwiązaniem jest użycie układów inercyjnych do stabilizacji lotu quadcopterów – opis przykładowo w pracy [18].

- **Analiza sportowa:**

Dzisiejszy sport wyczynowy osiąga powoli poziom do którego nie da się już dojść tylko odpowiednio dużo trenując i słuchając swojego ciała. Często treningi są poparte wieloma godzinami analiz pojedynczych ruchów i elementów. W pracy [19] został zamieszczony opis systemu do badania rzutu kulą na kręgielni.

2.3 Podsumowanie rozdziału

W tym rozdziale przedstawiono kilka najbardziej popularnych podejść do tematu nawigacji. Pokazane zostały ich główne zalety i wady. Szerzej omówiono układy inercyjne i ich roli w tym zagadnieniu. Z powodu swoich wad (opisanych w rozdziale czwartym) nie pozwalają one na dłuższe lokalizowanie, ale w krótkiej perspektywie, albo w połączeniu z innym systemem (np. GPS) doskonale spełniają swoją rolę.

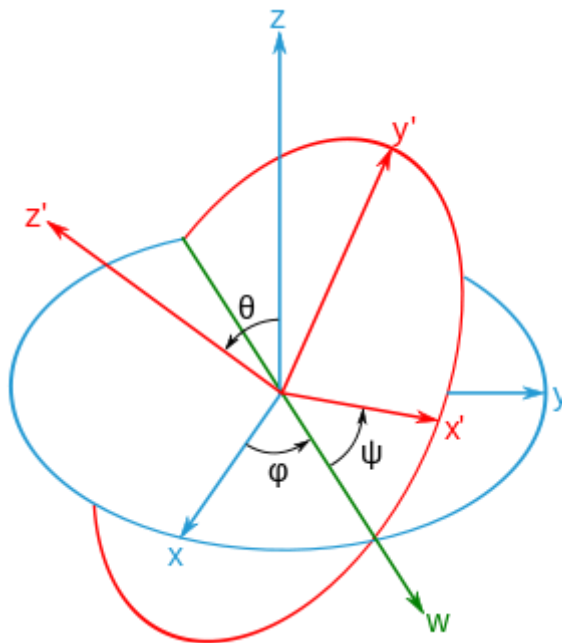
3 Opis teoretyczny

W tym rozdziale przedstawiono aparat matematyczny potrzebny do opisu i zrozumienia zagadnienia estymacji orientacji i położenia. Na początku zaprezentowano podejścia do opisu orientacji w przestrzeni, następnie skupiono się na szczególnego rodzaju filtrach wykorzystanych w fuzji czujników. W trakcie testów i implementacji postanowiono wprowadzić kilka modyfikacji, co stanowi temat kolejnego podrozdziału. Na samym końcu skupiono się na kalibracji jaka była niezbędna do rozpoczęcia.

3.1 Sposoby opisu orientacji

Orientacja jest to ułożenie obiektu w przestrzeni. Zazwyczaj określa się ją poprzez podanie zmiany względem ustalonego układu bazowego. Można ją opisywać na kilka sposobów. Każdy z nich cechuje inna charakterystyka.

- **Kąty Eulera**



Rysunek 4. Kąty Eulera. Źródło: [20]

Roll(φ) – kąt obrotu względem osi Ox ,

Pitch(θ) – kąt obrotu względem osi Oy ,

Yaw(ψ) - kąt obrotu względem osi Oz .

Zalety:

- Intuicyjne wartość, którą można sobie wyobrazić

Wady:

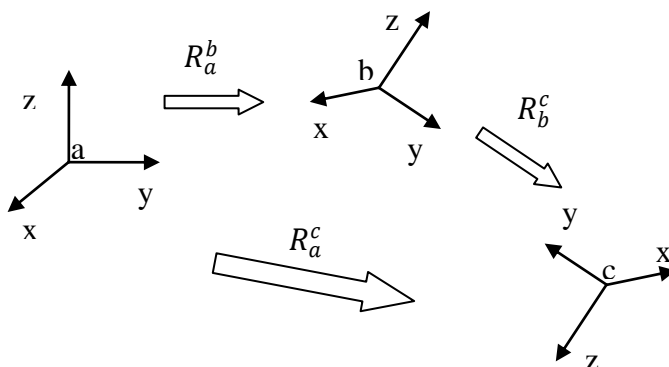
- Występowanie blokady (ang. gimbal lock), gdy dwie pierwsze osie znajdują się w jednej płaszczyźnie. Tak zwane ustawienia osobliwe i utrata jednego stopnia swobody. Wtedy zmieniając pierwszy kąt, równocześnie zmieniamy drugi.
- Konieczność ustalenia kolejności obrotów przy opisanu przekształcenia – możliwe 6 przypadków. Tutaj stosowana jest konwencja: XYZ.

- **Macierz rotacji - DCM** (ang. Direction Cosine Matrices)

Zależność pomiędzy kątami Eulera i macierzą rotacji.

Składanie pojedynczych obrotów w wypadkową macierz rotacji odbywa się poprzez mnożenie odpowiadającym za każdy obrót macierz.

$$R_a^c = R_a^b * R_b^c \quad (3.1)$$



Rysunek 5. Składanie rotacji. Źródło:[opracowanie własne]

Wprowadzenie zależności pomiędzy kątami Eulera polega na złożeniu obrotów względem osi Ox, Oy, Oz.

Wady:

- Nadmiarowa ilość zmiennych – do opisanie 3 obrotów używa się 9 wartości.
- Duża złożoność obliczeniowa,
- Kolejne rotacje nie dodają się – trzeba mnożyć macierz rotacji przez kolejną macierz odpowiadającą nowemu obrotowi.

Zalety:

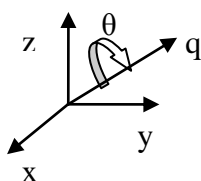
- Wiersze i/lub kolumny stanowią ortonormalne wektory (wersowy) bazowe konkretnych układów odniesienia.
- Jako konsekwencja powyższego punktu macierz jest nieosobliwa,

- **Kwaterniony**

$$Q_a^b = [q_0, q_1, q_2, q_3]^T \in R^4$$

$$q_0, q_1, q_2, q_3 \in R^1 \quad (3.2)$$

Interpretacja intuicyjna w konwencji oś i obrót - obrót wokół wektora jednostkowego q o kąt θ :



Rysunek 6. Kwaternion jako wektor i kąt obrotu wokół niego. Źródło: [opracowanie własne]

$$Q_a^b = [q_0, q_1, q_2, q_3]^T = \left[\cos \frac{\theta}{2}, x \sin \frac{\theta}{2}, y \sin \frac{\theta}{2}, z \sin \frac{\theta}{2} \right]^T \quad (3.3)$$

Q_a^b – orientacja układu a, względem układu b

Indeks dolny mówi jaki układ (ramka) jest opisywany, indeks górny – względem jakiego układu odniesienia (w jakim układzie) jest opisany..

Podstawowe operacje na kwaternionach:

Normalizacja:

$$|Q| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \quad (3.4)$$

$$Q_N = \frac{Q}{|Q|}$$

Jest ona niezbędna ponieważ w opisie orientacji występują 4 zmienne, co wprowadza nadmiarowość, czyli jedną orientację opisuje nieskończenie wiele kwaternionów. Dlatego też dla umożliwienia dalszych jednoznacznych obliczeń wykonywana jest normalizacja.

Pozwala ona także na znaczne uproszczenie i przyspieszenie obliczeń, ponieważ odwrotność kwaternionu znormalizowanego jest równa jego sprzężeniu.

Sprzężenie:

$$\overline{Q_a^b} = Q_b^a = [q_0, -q_1, -q_2, -q_3]^T \quad (3.5)$$

Pozwala zmienić nawzajem układy odniesienia w jakim są definiowane.

Składanie rotacji – q - kwaternion rotacji na początku o q_2 a następnie o q_1 (kolejność od prawej do lewej).

$$\begin{aligned} q_1 &= [x, y, z, w] \\ q_2 &= [x, y, z, w] \\ q &= q_1 q_2 \end{aligned} \quad (3.6)$$

Rotacja wektora $P = [xp, yp, zp]$ o kwaternion q_a^b :

$$\begin{aligned} q_p &= [xp, yp, zp, 0] \\ q_p' &= q_a^b q_p \overline{q_a^b} \end{aligned} \quad (3.7)$$

Mnożenie kwaternionów nie jest przemienne (obróć w odwrotnej kolejności skutkuje inną orientacją obiektu).

Wzory pozwalające przeliczyć kwaterniony na kąty Eulera:

$$\begin{aligned} roll &= \tan^{-1} \left(\frac{2 * (q_0 * q_1 + q_2 * q_3)}{1 - 2 * (q_1^2 + q_2^2)} \right) \\ pitch &= \sin^{-1} (2 * (q_0 * q_2 - q_3 * q_1)) \\ yaw &= \tan^{-1} \left(\frac{2 * (q_0 * q_3 + q_1 * q_2)}{1 - 2 * (q_2^2 + q_3^2)} \right) \end{aligned} \quad (3.8)$$

Początek wyprowadzenia kwaternionów na podstawie kątów Eulera:

Polega to na zapisaniu 3 kwaternionów obrotów zgodnych z kątami Eulera i następnie przemnożeniu nich w celu uzyskania złożenia tych obrotów.

Tworzymy trzy kwaterniony dla każdego z kątów Eulera:

$$\begin{aligned} q_{roll} &= \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \cos\left(\frac{roll}{2}\right) + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \sin\left(\frac{roll}{2}\right) \\ q_{pitch} &= \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \cos\left(\frac{pitch}{2}\right) + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \sin\left(\frac{pitch}{2}\right) \\ q_{yaw} &= \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \cos\left(\frac{yaw}{2}\right) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \sin\left(\frac{yaw}{2}\right) \end{aligned} \quad (3.9)$$

Następnie składamy kolejne obroty (wymnażamy) zgodnie z ustaloną kolejnością:

$$q = q_{roll} * q_{pitch} * q_{yaw} \quad (3.10)$$

Ostatnim etapem jest normalizacja wektora q otrzymanego w (3.10).

Wady:

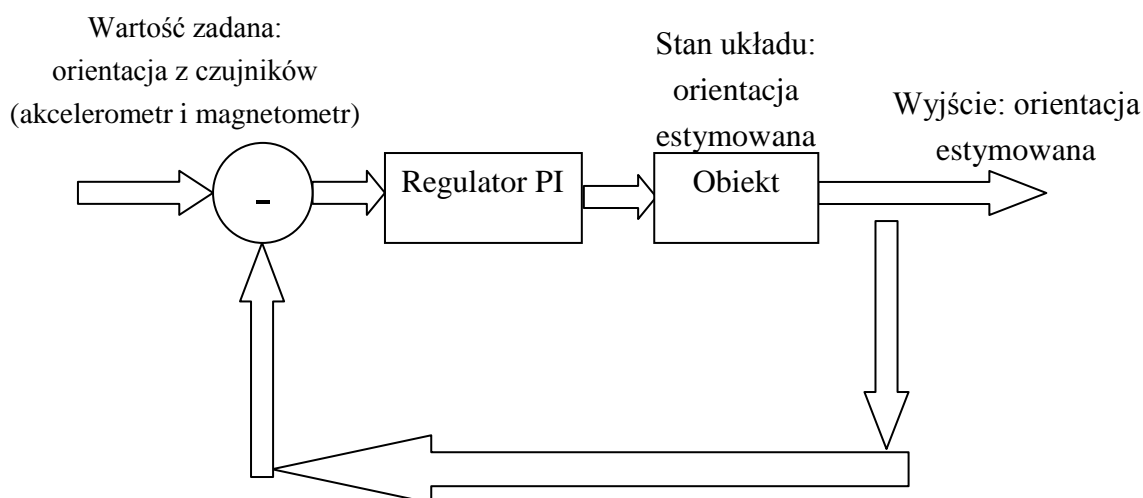
- Brak intuicyjnej informacji od razu z wartości,

Zalety:

- Szybkie obliczenia,
- Brak konfiguracji osobliwych,
- Łatwa interpolacja i ekstrapolacja – czyli znajdowanie pośredniego obrotu, albo predykcja dalszego.

3.2 Filtry

- **Filtr 1 - Mahony** – oparty na kwaternionach oraz z wykorzystujący regulator PI do kompensacji błędów powstałych przez dryft żyroskopu. W celu zlikwidowania odchyłki od pionu wykorzystuje się akcelerometr, od poziomu – kierunku północnego magnetometr. Z powodu wykorzystywanych czujników nazywa się taki system MARG (ang. magnetic, angular rate and gravity). [21]



Rysunek 7. Zamknięty układ regulacji zastosowany w systemie MARG.

Przyjęto, że układ globalny odniesienia jest połączony z ziemią – punktem startowym czujnika. Zwrot osi OX wskazuje na północ(ang. north), OY na wschód(ang. east), OZ na górę(ang. up) – układ NED. Dla uproszczenia ominięto kulistość ziemi.

Lokalnym układem odniesienia jest układ powiązany z płytą skaloną (samolotem).

Kwaternion rotacji (przejścia) z układu globalnego do lokalnego oznaczono jako

$$\begin{aligned} Q_{local}^{global} &= [q_0, q_1, q_2, q_3]^T \\ q_0 &- \text{część dotycząca kątu obrotu} \\ [q_1, q_2, q_3]^T &- \text{część dotycząca osi wokół której obrót} \end{aligned} \quad (3.11)$$

Pozwala on na uzyskanie w jakiej orientacji jest aktualnie samolot, co jest niezbędne do poprawnego sterownia, czy choćby wizualizacji stanu.

Przed wykonaniem obliczeń dokonywana jest normalizacja wartości z pomiarów przyspieszenia (a) i pola grawitacyjnego(m). Oznaczono je kolejno (a_N) i (m_N). Dlaczego jest to tak ważne zostało opisane w wstępie teoretycznym.

Magnetometr wyznacza kierunek północny w globalnym układzie odniesienia. Wykorzystując poniższe wzory pomocnicze:

$$\begin{aligned} hx &= 2 \left[m_{Nx} \left(\frac{1}{2} - q_2^2 - q_3^2 \right) + m_{Ny} (q_1 q_2 - q_0 q_3) \right. \\ &\quad \left. + m_{Nz} (q_1 q_3 + q_0 + q_2) \right] \\ hy &= 2 \left[m_{Nx} (q_1 q_2 + q_0 q_3) + m_{Ny} \left(\frac{1}{2} - q_1^2 - q_3^2 \right) \right. \\ &\quad \left. + m_{Nz} (q_2 q_3 - q_0 + q_1) \right] \\ hz &= 2 \left[m_{Nx} (q_1 q_3 - q_0 q_2) + m_{Ny} (q_2 q_3 - q_0 q_1) \right. \\ &\quad \left. + m_{Nz} \left(\frac{1}{2} - q_1^2 - q_2^2 \right) \right] \\ bxy &= \sqrt{hx^2 + hy^2} \\ bz &= hz \end{aligned} \quad (3.12)$$

Obliczamy prędkość kątową doregulowania układu bazowego do kierunku północnego.

$$\begin{aligned} wx &= 2 \left[bxy \left(\frac{1}{2} - q_2^2 - q_3^2 \right) + bz (q_1 q_3 - q_0 q_2) \right] \\ wy &= 2 [bxy (q_1 q_2 - q_0 q_3) + bz (q_0 q_1 + q_2 q_3)] \\ wz &= 2 \left[bxy (q_0 q_2 + q_1 q_3) + bz \left(\frac{1}{2} - q_1^2 - q_2^2 \right) \right] \end{aligned} \quad (3.13)$$

Analogicznie wykonujemy z przyspieszeniem grawitacyjnym, który wyznacza prędkość kątową doregulowania układu bazowego do pionu. Zmienne pomocnicze:

$$\begin{aligned} vx &= 2(q_1 q_3 - q_0 q_2) \\ vy &= 2(q_0 q_1 + q_2 + q_3) \\ vz &= q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{aligned} \quad (3.14)$$

Obliczono sumaryczny błąd (uchyby regulacji). Jednostka uchybu to kąt w radianach.

$$\begin{aligned} e_x &= (a_{Ny} vz - a_{Nz} vy) + (m_{Ny} wz - m_{Nz} wy) \\ e_y &= (a_{Nz} vx - a_{Nx} vz) + (m_{Nz} wx - m_{Nx} wz) \\ e_z &= (a_{Nx} vy - a_{Ny} vx) + (m_{Nx} wy - m_{Ny} wx) \end{aligned} \quad (3.15)$$

Obliczenie wypadkowej prędkości kątowej po uwzględnieniu wartości sterowania z regulatora PI:

$$\begin{aligned} g_{xfinal} &= g_x + u_x \\ g_{yfinal} &= g_y + u_y \\ g_{zfinal} &= g_z + u_z \end{aligned} \quad (3.16)$$

Gdzie:

$$\begin{aligned} u_x &= Kpe_x + Ki \sum e_x \\ u_y &= Kpe_y + Ki \sum e_y \\ u_z &= Kpe_z + Ki \sum e_z \end{aligned} \quad (3.17)$$

Wartości te zostały użyte w celu aktualizacji orientacji (stanu systemu)

$$\begin{aligned} q_w &+= \frac{(-q_1 g_{xfinal} - q_2 g_{yfinal} - q_3 g_{zfinal}) T_{sample}}{2} \\ q_x &+= \frac{(q_0 g_{xfinal} + q_2 g_{zfinal} - q_3 g_{yfinal}) T_{sample}}{2} \\ q_y &+= \frac{(q_0 g_{yfinal} + q_3 g_{xfinal} - q_2 g_{zfinal}) T_{sample}}{2} \\ q_z &+= \frac{(q_0 g_{zfinal} + q_1 g_{yfinal} - q_2 g_{xfinal}) T_{sample}}{2} \end{aligned} \quad (3.18)$$

Podobnie jak wcześniej – otrzymane wartości końcowe zostały poddane normalizacji.

- **Filtr 2 - Madgwick** – oparty na kwaternionach, wykorzystuje metodę optymalizacji najszybszego spadku. Z opracowaniem teoretycznym [22] autor zamieścił również przykładową implementację w języku C.
- **Kalmana** - Dokładny opis, wyprowadzenie i przykładową implementację znajdziemy w [23]. Istnieje też polskie opracowanie [24], gdzie autor porównuje dodatkowo zastosowanie 3 filtrów dla każdego z kątów Eulera w formie poprawnej teoretycznie (bez wejścia sterowania) oraz również poprawnie działającej, ale nie zgodnej z założeniami poczynionymi podczas wyprowadzenia (wg. poprawnego założenia sterowanie powinno być deterministycznym sygnałem) filtru Kalmana (z wejściem sterującym, które jest niedeterministycznym sygnałem – odczyt z czujników fizycznych).
- **Komplementarny** – prosty i popularnym przykładem filtru nieliniowego. Pozwala on połączyć sygnały, które posiadają zakłócenia w różnych pasmach częstotliwościowych.

W zastosowaniu dla określania orientacji na podstawie akcelerometru i żyroskopu można go określić poniższym dyskretnym wzorem rekurencyjnym:

$$\hat{\theta}_{n+1} = \alpha * \theta_{acc_n} + (1 - \alpha) * (\hat{\theta}_n + \omega_{gyro_n} * dt) \quad (3.19)$$

Gdzie: $\alpha \in \langle 0, 1 \rangle$ jest wagą. Im α jest bliższa 1 tym bardziej na nową wartość wpływa odczyt z akcelerometru, gdy α się zmniejsza to z żyroskopu. Przy małym dryfcie żyroskopu tym mniejszą α można przyjąć i tym stabilniejszy będzie odczyt.

• Podsumowanie:

Ponieważ platforma na której oparty jest ten projekt jest wystarczająco szybka i posiada dodatkowo koprocesor do obliczeń zmiennopozycyjnych zdecydowano się na wybranie

algorytmu pierwszego. Spowodowane było to również jego ogólną dostępną implementacją, oraz dobrą dokumentacją. Zagadnienie filtracji nie kończy się jednak na wyżej wymienionych podejściach. Istnieje bardzo dużo sposobów nieliniowego filtrowania sygnałów jednak często opracowywanie ich i badanie ich charakterystyki mogłoby stanowić podstawy do napisania kolejnej pracy.

3.3 Wprowadzone modyfikacje

- **Uproszczenie modelu ziemi**

Z powodu dużej częstotliwości odświeżania pozycji, co skutkuje małymi wartościami zmian położenia w kolejnym kroku, postanowiono pominąć krzywiznę ziemi przyjmując, że jest ona płaska w bliskim otoczeniu. Przeliczanie przemieszczeń na współrzędne (opisane np. według standardu WGS84) mogło by skutkować w dużych błędach numerycznych spowodowanych dość złożonymi wzorami. W Internecie są dostępne implementacje bibliotek np.: [25], które to umożliwiają. W dalszych etapach rozwoju projektu możliwe będzie szybkie wprowadzenie ewentualnie tego elementu ponownie. Jednak, ponieważ przyrosty są maksymalnie rzędu kilkuset centymetrów to przyjęte uproszczenie jest w pełni wystarczające.

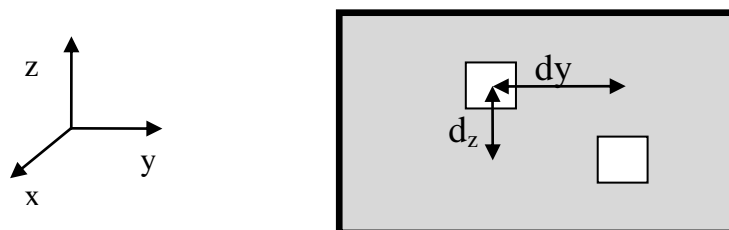
- **Obroty i przyspieszenie odśrodkowe**

Podczas obrotów na akcelerometr działa przyspieszenie odśrodkowe, która wyraża się wzorem:

$$a = \omega^2 r \quad (3.20)$$

By usunąć ten wpływ i uzyskać czyste przyspieszenie liniowe od wartości zmierzonego przyspieszenia odjęto składową pochodzącą z tego źródła.

Ułożenie czujników na płycie przedstawia się następująco:



Rysunek 8. Ułożenie sensorów na płycie pomiarowej.

Tabela 2. Odległość pomiędzy środkami czujników wzdłuż kolejnych osi:

Nazwa	Długość, mm:
d_x	5
d_y	3
d_z	0

Dla kolejnych przyspieszeń wokół:

- Osi Ox :

$$a_x = \omega_x^2 d_x \quad (3.21)$$

rozkładając na składowe:

$$a_{xx} = 0 \quad (3.22)$$

$$a_{xy} = a_x \quad (3.23)$$

$$a_{xz} = 0 \quad (3.24)$$

○ Osi Oy:

$$a_y = \omega_y^2 d_y \quad (3.25)$$

rozkładając na składowe:

$$a_{yx} = a_y \quad (3.26)$$

$$a_{yy} = 0 \quad (3.27)$$

$$a_{yz} = 0 \quad (3.28)$$

○ Osi Oz:

$$a_z = \omega_z^2 r_z \quad (3.29)$$

$$r_z = \sqrt{d_x^2 + d_y^2}$$

rozkładając na składowe:

$$a_{zx} = a_z \sin \alpha \quad (3.30)$$

$$\sin \alpha = \frac{d_y}{\sqrt{d_x^2 + d_y^2}}$$

$$a_{zy} = a_z \cos \alpha \quad (3.31)$$

$$\cos \alpha = \frac{d_x}{\sqrt{d_x^2 + d_y^2}}$$

$$a_{zz} = 0 \quad (3.32)$$

Odjęcie od wartości zmierzonych składowych wywołanych przez siłę odśrodkową:

$$\begin{aligned} a_x' &= a_{x_zmierzone} - (a_{xx} + a_{yx} + a_{zx}) \\ a_y' &= a_{y_zmierzone} - (a_{xy} + a_{yy} + a_{zy}) \\ a_z' &= a_{z_zmierzone} - (a_{xz} + a_{yz} + a_{zz}) \end{aligned} \quad (3.33)$$

Dodatkowo wprowadzono minimalną wartość progową, jaką musi osiągnąć prędkość kątowa, by uwzględnić jej wpływ poprzez powyższy wzór 3.33.

• Szybkie obroty

W trakcie testów zauważono przy szybkich i dynamicznych obrotach filtr czasem przestawał się stabilizować. Po badaniu problemu i poszukiwaniach rozwiązania okazało się, że gdy część całkująca regulatora osiągnęła pewną wartość a tym samym sterowanie stało się za duże to stało się niemożliwe osiągnięcie ponownie aktualnej orientacji, albo wymagało to długiego czasu. Jest tak z powodu, że wartość kątów ma ograniczoną wartość, która jest okresowa np. $(-180^\circ; +180^\circ)$.

Problem ten został opisany również dokładniej w [26]. Opierając się na tych analizach wprowadzono modyfikację polegającą na zwiększaniu wzmocnienia regulatora PI, gdy wykryto szybki obrót (poprzez monitorowanie wartości żyroskopu).

Zdarza się, że zwiększanie wzmocnienia w regulatorze tego typu skutkuje utratą stabilności. Opierając się jednak na paru doświadczeniach na zgromadzonych danych i podobnym badaniu w wcześniej wspomnianej pracy zauważono, że system pozostaje stabilny przy takich zmianach.

3.4 Kalibracja

1. Czujniki - Akcelerometr

Sygnałem wzorcowym było przyspieszenie ziemskie o znanej wartości ($g = 9.81 \text{ m/s}^2$). Zmierzono przyspieszenie $+g$ i $-g$ dla każdej z osi. Precyzyjne ustalenie poziomu odbywało się przy pomocy poziomiccy cyfrowej. Pomiar został wykonany wielokrotnie, a wynik końcowy uśredniono. Obliczono przesunięcie zerowego przyspieszenia i współczynnik przeliczający wartości z czujnika na jednostki SI (m/s^2).

$$acceleration = gain * (rawAcceleration - offset) \quad (3.34)$$

Przykładowo dla osi OX:

$$offset_x = \frac{\sum_{i=1}^n accX_i^+ + \sum_{i=1}^n accX_i^-}{2 * n} \quad (3.35)$$

$$gain_x = \frac{\sum_{i=1}^n accX_i^+ - \sum_{i=1}^n accX_i^-}{2 * n * 9.81} \quad (3.36)$$

gdzie: $accX_i^+$ - wartość i - tego pomiaru na osi OX przy przyspieszeniu $+g$.
 n – ilość pomiarów.

Dodatkowo zaimplementowana została procedura precyzyjnego ustalania współczynnika wzmocnienia w trakcie działania. Oblicza on średnią długość wektora i oblicza stosunek do teoretycznej wartości.

$$gain_{correction} = \frac{\sum_{i=1}^n \sqrt{accX_i^2 + accY_i^2 + accZ_i^2}}{9.81} \quad (3.37)$$

2. Czujniki - Żyroskop

Podobnie jak w akcelerometrze obliczono offsetu i wzmocnienia. Przesunięcie zera zostało obliczone poprzez uśrednienie wartości z pomiarów, gdy czujnik pozostawał nieruchomy. Współczynnik wzmocnienia został obliczony poprzez scałkowanie prędkości kątowej przy obrocie względem jednej osi o znany kąt i porównanie obydwu wartości. Zostało to wykonane dla kąta 180° i 360°

$$gain_x = \frac{\sum_{i=1}^n gyroX_i * dt}{znanyKąt} \quad (3.38)$$

gdzie: $gyroX_i$ - wartość prędkości kątowej wokół osi OX w i - tym pomiarze.
 n – ilość pomiarów.
 dt – czas próbkowania $\frac{1}{400} \text{ Hz} = 2.5 \text{ ms}$

3. Czujniki – Magnetometr

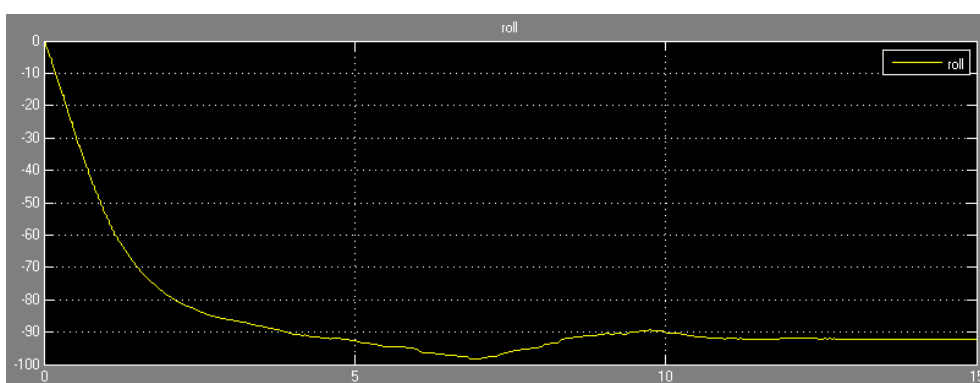
Kalibracja magnetometru składa się z dwóch etapów. Pierwszy to kalibracja tzw. hard-iron, która kompensuje przesunięcie wartości spowodowane elementami które znajdują się na płytce PCB oraz w okolicy czujnika. Procedura jest zbliżona do tej z akcelerometru. Znajdujemy maksymalną i minimalną wartość mierzoną na każdej z osi i obliczamy offset. Ponieważ wartość pola magnetycznego nie jest wrażliwa na ruch to procedura polegała na obracaniu czujnika w wszystkich kierunkach przez zadany czas i wyznaczeniu ekstremalnych

wartości. Następny etap to kalibracja soft-iron. Bez niej wartości mierzonego pola nie tworzą jednorodnego obszaru (kuli) tylko elipsę co powoduje niejednorodny odczyt wartości. Procedura polega na wyznaczeniu np. metodą najmniejszych kwadratów macierzy wzmocnienia [27].

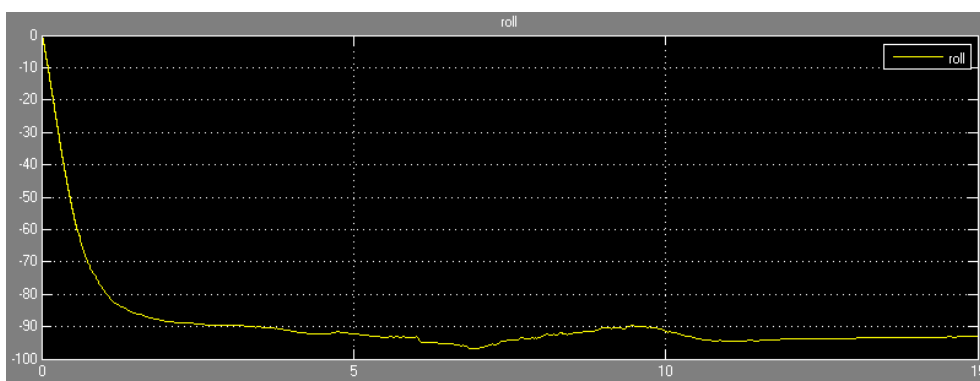
4. Filtry:

Kalibracja odbyła się na podstawie badania odpowiedzi filtru na skok jednostkowy. Przy zadanych początkowych estymowanych kątach obrócono czujnik do położenia $[92^\circ, 0^\circ, 0^\circ]$ (kąty [roll, pitch, yaw]) i uruchomiono filtrację. W doborze parametrów wzmocnienia i współczynnika całkowania kierowano się by uzyskać stosunkowo szybki czas stabilizacji przy możliwie małych oscylacjach i przeregulowaniach.

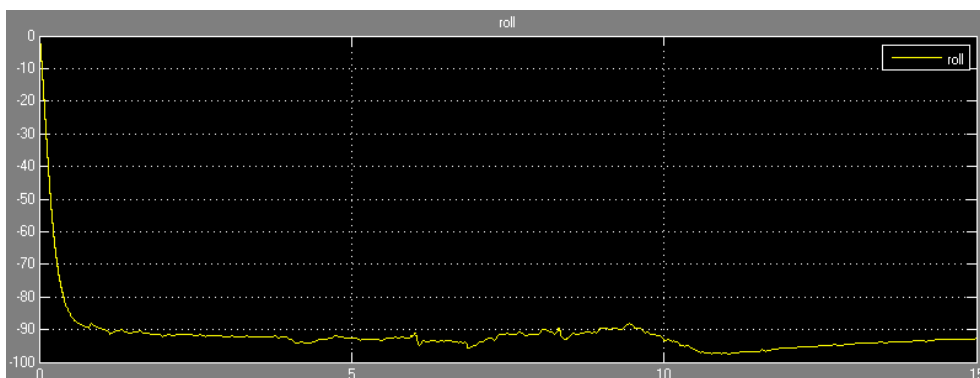
Oś rzędnych (OY) przedstawia wartość kąta w stopniach, oś odciętych (OX) czas w sekundach.



Rysunek 9. Odpowiedź filtru ($K_p = 1$, $K_i = 0$).



Rysunek 10. Odpowiedź filtru ($K_p = 2$, $K_i = 0$).



Rysunek 11. Odpowiedź filtru ($K_p = 5$, $K_i = 0.7$).

Analogiczna procedura została wykonana dla pozostałych implementacji filtrów. W trakcie testów pracy online zauważono jednak, że parametry wyznaczone przy wolnych obrotach są nieodpowiednie przy dynamicznych ruchach. Zbyt duży czas całkowania, powodował dobrą stabilizację i niwelowanie uchybu ustalonego w przypadku wolnych obrotów, przy szybkich jednak potrafił bardzo zniekształcać wartości, oraz czasem zbyt duże zmiany kumulowały się, co powodowało, że układ bardzo długo się stabilizował, lub nawet stawał się oscylacyjny (niestabilny).

3.5 Podsumowanie rozdziału

Jak łatwo zauważyć istnieje kilka podejść do opisywania orientacji, każde z nich posiada obszary w jakich można je stosować i w jakich powodują one pewne problemy. Kwaterniony są często stosowane w gdzie liczy się szybkość działania i nie potrzeba intuicyjnej interpretacji wartości. Macierze rotacji stosowane są np. w amatorskich algorytmach, gdzie każda z kolumn i wierszy stanowi szybką informację o rozkładzie wektora w układzie odniesienia. Kąty Eulera natomiast stanowią dobrą alternatywę, gdy ważny jest szybki podgląd wartości oraz występujące konfiguracje osobliwe nie stanowią problemu. Przedstawione podejścia do estymacji orientacji pomimo złożonych podstaw teoretycznych w implementacji właściwej sprowadzają się do kilkudziesięciu działań. W trakcie testowania wykryto pewne szczególne momenty – szybkie rotacje, oraz wpływ siły odśrodkowej – które potrzebowały wprowadzenia dodatkowych obliczeń i korekt.

4 Opis użytych technologii

4.1 Sprzęt

1. STM32M4f Discovery - Cortex M4F:

Wykorzystana płytką zawiera 32-bitowy mikrokontroler firmy ARM z rdzeniem Cortex-M4F, 1 MB pamięci Flash, 192 kB RAM. Dodatkowo do procesora podłączony jest koprocesor zmiennoprzecinkowy, który znacznie zmniejsza czas obliczeń. Procesor posiada też specjalne zoptymalizowane funkcje DSP np. szybkie obliczanie funkcji matematycznych (sin, cos, sqrt), filtrowanie cyfrowe (IIR, FIR), transformaty (FFT, DCT) oraz operacje na macierzach.

2. Czujniki:

- **Akcelerometr** [28]:

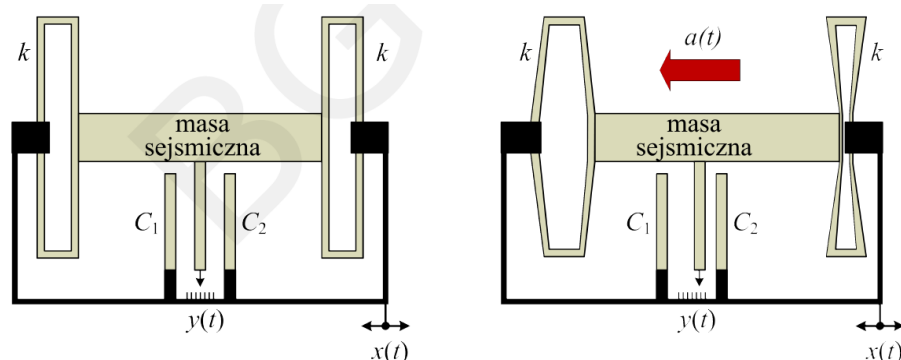
Czujnik pozwala na pomiar przyspieszenia działającego na dane urządzenie. Można wyróżnić następujące tryby pracy.

Akcelerometry tensometryczne - na belce zamocowane jest masa, która pod wpływem przyspieszenia ugina belkę. Zamieszczone tensometry mierzą tą zmianę i pozwalają przeliczyć ją na przyspieszenie. Takie czujniki posiadają stosunkowo niskie częstotliwościowe pasmo przenoszenia, oraz umożliwiają pomiar stałego w czasie przyspieszenia.

Wibrometry - zbudowane są z wiotkiej sprężyny, o małym współczynniku sprężystości i dużej masy na niej zaczepionej. Umożliwia pomiar niskich częstotliwości do kilkunastu Hz.

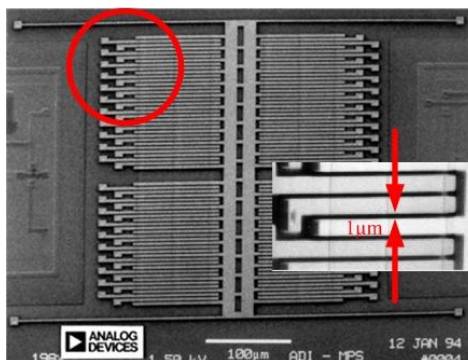
Czujniki piezoelektryczne - ich zasada działania polega na pomiarze ładunku na ścianach kryształu kwarcu pod wpływem sił ściskających lub rozciągających od przyczepionej do niego masy. Ten rodzaj pomiaru cechuje bardzo szerokie pasmo częstotliwościowe i mały współczynnik tłumienia.

Akcelerometry pojemnościowe - zbudowane są z dwóch kondensatorów, które mają jedną wspólną płytkę. Przyczepiona jest do niej masa na sprężynach. Gdy na urządzenie działa przyspieszenie to płytki wychyla się z położenia równowagi, a tym samym zmienia się pojemność dwóch kondensatorów C_1 i C_2 . Układ elektroniczny działając w pętli sprzężenia zwrotnego, tak dobiera napięcie na centralnej płytce, by utrzymać ją w położeniu zerowym. Wartość napięcia podawanego na tę ramkę jest miarą działającego przyspieszenia.



Rysunek 12. Akcelerometr pojemnościowy, Źródło: [28]

Ten rodzaj jest zazwyczaj stosowany w układach wbudowanych typu MEMS. W układach wykonywane jest kilkadziesiąt warstw takich czujników, co zostało pokazane w powiększeniu na poniższym rysunku.



Rysunek 13. Przybliżenie układu MEMS, Źródło: [29]

Główne źródła błędnych pomiarów z akcelerometrów typu MEMS to:

- Stałe przesunięcie (ang. bias) - może być ustalone w czasie kalibracji, gdy po ustawieniu czujnika w znanym położeniu odejmiemy składową grawitacji (większy opis w kalibracji). Błąd ten podczas całkowania zwiększa błąd położenia z kwadratem czasu.
- Losowy szum biały (ang. velocity random walk), który poprzez wielokrotne sumowanie powoduje zwiększanie wariancji sygnału z biegiem czasu (proporcjonalny do pierwiastka z numeru próbek),
- Zmiana przesunięcia (ang. bias) pod względem temperatury – zależność zazwyczaj nieliniowa.
- Błędy kalibracji – zły współczynnik wzmocnienia lub przesunięcia offsetu.

W pracy został wykorzystany akcelerometr Analog Devices ADXL345 [30].

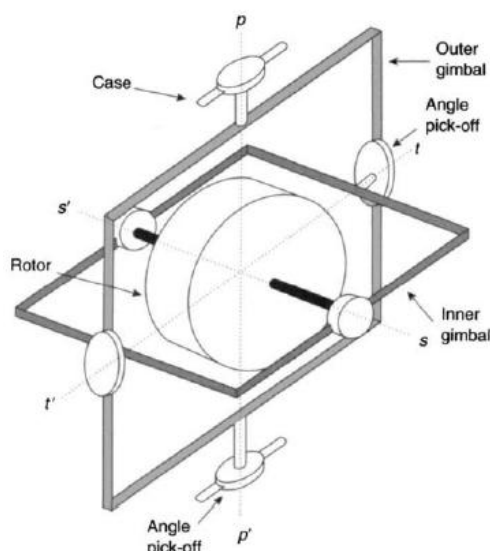
Tabela 3. Główne właściwości akcelerometru ADXL345

Zakres pracy	$\pm 2g, \pm 4g, \pm 8g, \pm 16g$
Rozdzielczość	3.9mg/LSB dla $\pm 2g$, 7.8mg/LSB dla $\pm 4g$
Częstotliwość pomiaru	6,25-3200 HZ
Komunikacja	SPI/I ² C

• Żyroskop:

Mierzy prędkość kątową z jaką obracane jest urządzenie lub bezpośrednio jego orientację. W tej dziedzinie można wyróżnić kilka klas urządzeń np.: mechaniczne, mikro-elektromechaniczne systemy (MEMS), optyczne (światłowodowe FOG, laserowe z komorą wyładowczą RLG), elektrostatyczne (ESG), wibracyjne.

Żyroskopy mechaniczne - wykorzystuje zasadę zachowania momentu pędu wirującej masy zawartej w ramkach. Zgodnie z teorią, gdy na układ nie działają inne momenty sił wartość prędkości kątowej pozostają bez zmian. Co za tym idzie, gdy obiekt zmienia swoje położenie masa zachowuje swoje pierwotne ułożenie. Mierząc kąty wychylenia ramki (w osiach pp' , tt' , ss' na rysunku 6), uzyskujemy bezpośrednio orientację.



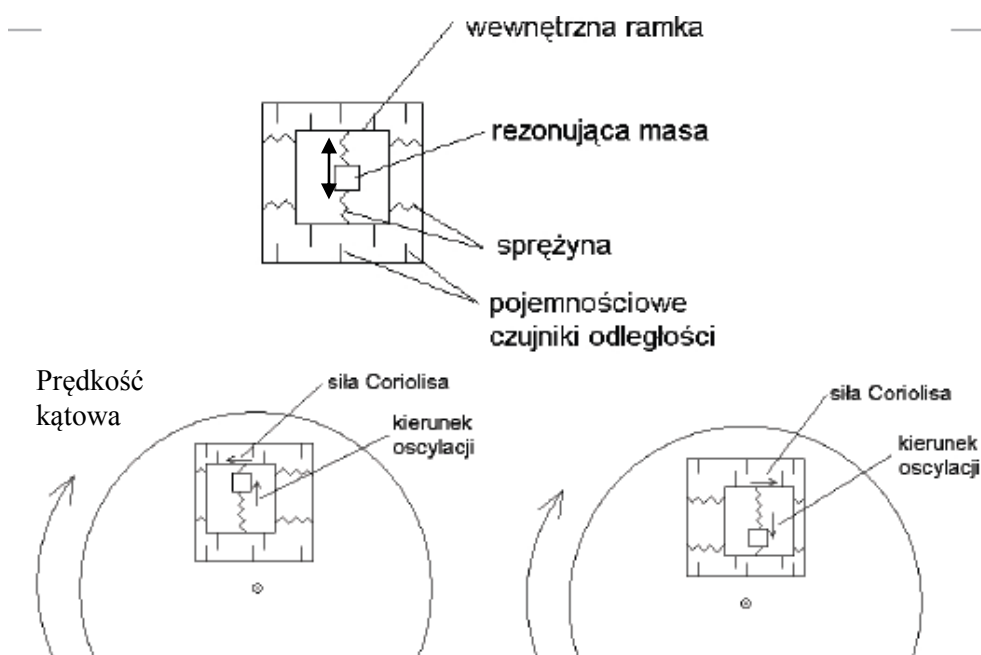
Rysunek 14. Żyroskop mechaniczny, Źródło: [20]

Rozwiązanie mechaniczne jest jednak z powodu wielu ruchomych części, sporych rozmiarów i konieczności podtrzymywania obrotu masy nie stosowane w systemach nawigacji. Praktycznie w zastosowaniach do tworzenia systemów inercyjnych IMU używane są żyroskopy optyczne i ostatnio coraz częściej typu MEMS.

Żyroskopy optyczne – wykorzystuje zjawiska Sagnaca – polega ono na różnicowaniu czasu przebycia drogi optycznej u układzie poruszającym się z prędkością kątową.

Żyroskopy MEMS - pomiar prędkości kątowej opiera się o zasadę działania siły Coriolisa. Intuicyjnie opisuje ona jakie przyspieszenie musi być przyłożone do ciała, aby mogło utrzymać się na obracającej powierzchni.

$$\vec{F}_c = -2m(\vec{\omega} \times \vec{v}) \quad (4.1)$$



Rysunek 15. Wykorzystanie siły Coriolisa w żyroskopach MEMS, Źródło: [31]

Główne źródła błędów w żyroskopach typu MEMS to:

- Stałe przesunięcie pomiarów (bias) – może być skompensowane poprzez odejmowanie uśrednionej wartości z dłuższego czasu pomiaru.
- Losowy szum biały (ang. angle random walk), który poprzez wielokrotne sumowanie powoduje zwiększanie wariancji sygnału z biegiem czasu (proporcjonalny do pierwiastka z numeru próbek),
- Zmiana przesunięcia pod względem temperatury – zależność zazwyczaj nieliniowa.
- Błędy kalibracji – zły współczynnik wzmocnienia lub przesunięcia offsetu.

W układzie zastosowano czujnik ITG-3200 [32].

Tabela 4. Główne właściwości żyroskopu ITG3200

Zakres pracy	± 2000 °/s
Czułość	14,375 LSB/(°/s)
Częstotliwość pomiaru	5 – 256 HZ
RMS błędu	0.38 °/s
Komunikacja	I ² C

- **Magnetometr:**

Mierząc tylko natężenie pola ziemskiego możemy uzyskać kierunek jaki wskazuje urządzenie. Czujniki typu MEMS mogą działać na podstawie różnych zasad fizycznych np. efekt Halla, kwantowa natura elektronów (GMR), czy siła Lorentza. Więcej o każdym ze sposobów w [33].

Główne źródła błędów to:

- Stałe przesunięcie pomiarów (ang. bias) – analogicznie jak w żyroskopie,
- Zakłócenia z zewnętrznych pól magnetycznych np. baterii, silników, głośników, dużych prądów w ścieżkach i przewodach.

W układzie zastosowano czujnik HMC5883L [34].

Tabela 5. Główne właściwości magnetometru HMC5883L

Zakres pracy	± 8 gauss
Czułość	230 – 1370 LSB/(gauss)
Częstotliwość pomiaru	0,75 – 75 HZ (w trybie ciągłym)
Komunikacja	I ² C

Podsumowując przy wykorzystywaniu tych czujników do wyznaczania orientacji pomiary z akcelerometru są narażone na szumy wysoko częstotliwościowe. Wyznaczane kąty są bardzo wrażliwe na małe zaburzenia. Żyroskop oferuje stabilniejszy odczyt, ale jest narażony na dryft wartości z powodu sumowania pomiarów. Magnetometr natomiast jest bardzo wrażliwy na wszelkie lokalne źródła pola magnetycznego – przez co w okolicy silników, metali, czy nawet budynków jego odczyt ulega znaczącemu zniekształceniu.

4.2 Oprogramowanie

W trakcie pisania oprogramowania na procesor ARM wykorzystano język C++. Użyto darmowego środowiska Eclipse z skonfigurowanym debugger'em OpenOCD i kompilatorem GNU GCC ARM. Dokładna dokumentacja, przejrzysty opis konfiguracji środowiska i jego najważniejsze elementy znajdują się na stronie [35].

Do testów implementowanych algorytmów użyto środowiska Matlab i Simulink. Szczególnie przydatne okazały się bloczki udostępniane w pakiecie DSP System Toolbox. Dzięki nim w szybki sposób można obliczyć parametry statystyczne oraz wczytać zgromadzone próbki, zaprojektować filtr cyfrowy, czy wykonać obliczenia macierzowe.

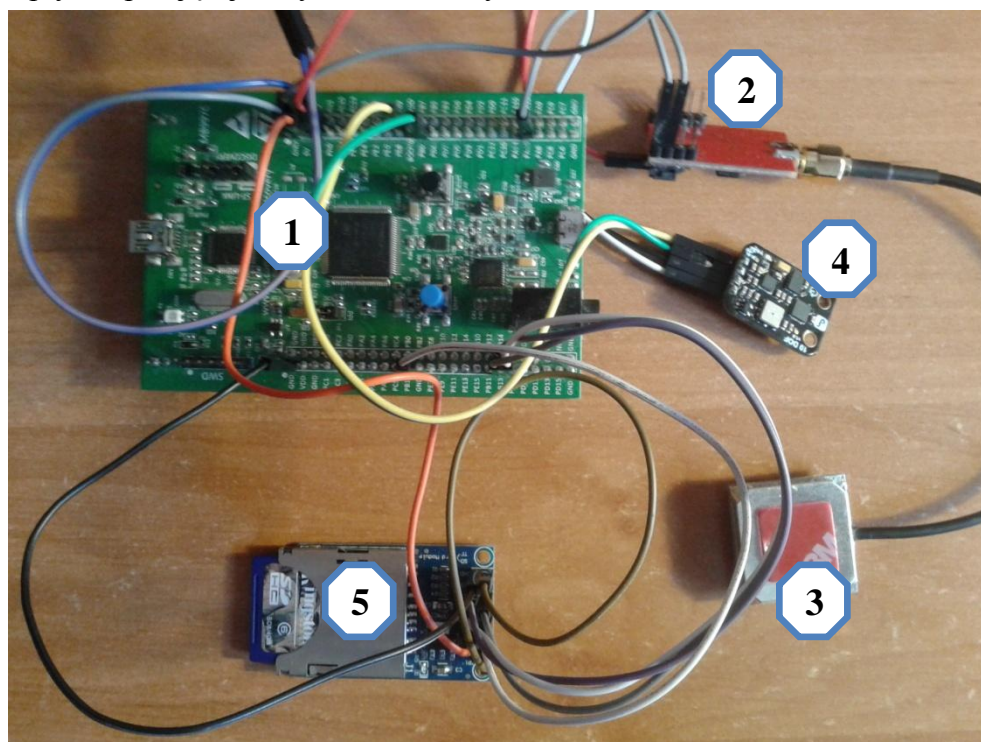
Ostatnim elementem było stworzenie graficznego interfejsu, który powstał w Microsoft Visual Studio 2013 w języku C++. Prosty graficzny kreator okienek i łatwe tworzenie obsługi przycisków spowodowało, że tworzenie i modyfikowanie GUI było stosunkowo szybkie.

5 Opis wyniku końcowego pracy

Poniższy rozdział może zostać podzielony na 4 główne podrozdziały. Pierwszy to ogólny opis wprowadzający. Następny - programy i skrypty powstałe w trakcie testów i prototypowania w programach Matlab i Simulink. W trzecim zawarte zostały informacje na temat programu głównego napisanego na platformie STM32 Discovery. Ostatni zajmują się tematem graficznego interfejsu użytkownika.

5.1 Elementy składowe

Główny czas został przeznaczony na implementację i oprogramowanie systemu dlatego też połączenie jest na etapie prototypowej za pomocą kabli. Do układu zaplanowane jest dołączenie kilku dodatkowych układów dlatego też postanowiono do momentu zdecydowania do których pinów zostaną one podłączone i zaimplementowania ich funkcjonalności nie projektować płytki spinającej wszystkie elementy.

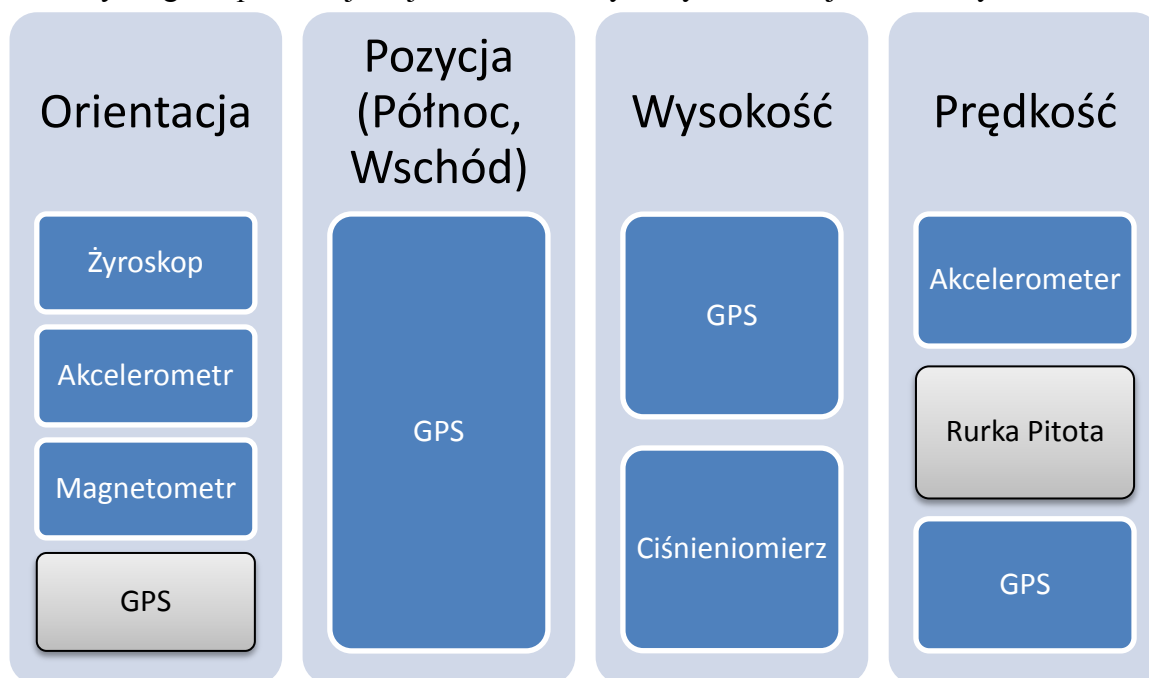


Rysunek 16. Prototyp systemu. Płytki STM32M3f Discovery (1), układ GPS (2), antena do GPS (3), układ IMU – czujników (4), adapter karty SD i translator napięć (5). Źródło: (opracowanie własne) .

Tabela 6. Fizyczne podłączenie elementów do płytki

Element:	Piny
Czujniki (SPI1)	SCL – GPIO_B6 SDA – GPIO_B9
GPS (USART1)	TX – GPIO_A9 RX – GPIO_A10
Karta SD (SPI2)	CS – GPIO_B1 SCK – GPIO_B13 MISO – GPIOB_14 MOSI – GPIOB_15

Poniższy diagram prezentuje w jakich celach wykorzystano kolejne elementy.

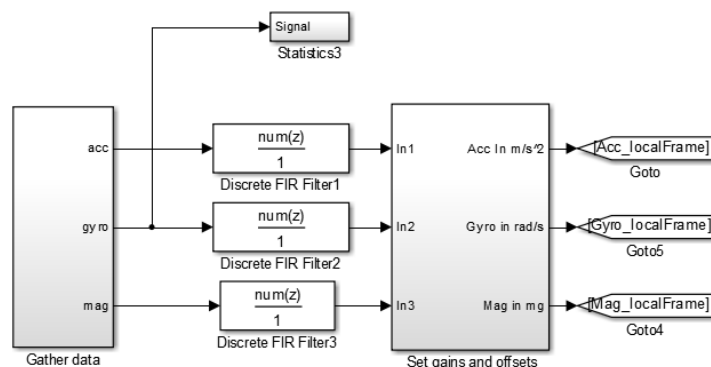


- Orientacja– fuzja czujników inercyjnych i GPS:
 - Algorytm fuzji czujników zwracający kwaternion przekształcenia z układu globalnego do lokalnego będący tożsamy z aktualną orientacją,
 - Adaptacyjny filtr komplementarny kąta yaw – gdy dane z GPS są niedostępne, lub prędkość poniżej progu waga jest zmieniana tak, by brać pod uwagę tylko wartość z układów inercyjnych. Po ponownym odbiorze sygnału GPS powrót do poprzedniej wartości. (w planach)
- Wysokość – fuzja czujnika ciśnienia i GPS:
 - Średnia ważona uwzględniająca aktualną wartość współczynnika HDOP GPS’a, pomiar z czujnika ciśnienia i poprzednią wartość.
- Prędkość:
 - Próba dodania estymacji prędkości z układów inercyjnych – całkowanie przyspieszenia mierzonego akcelerometrem w lokalnym układzie współrzędnych – obliczenie prędkości względem wiatru,
 - GPS – względem ziemi,
 - Dołączenie rurki Pitota w celu obliczenia dokładnej prędkości względem wiatru. (w planach)

5.2 Fragmenty implementacji

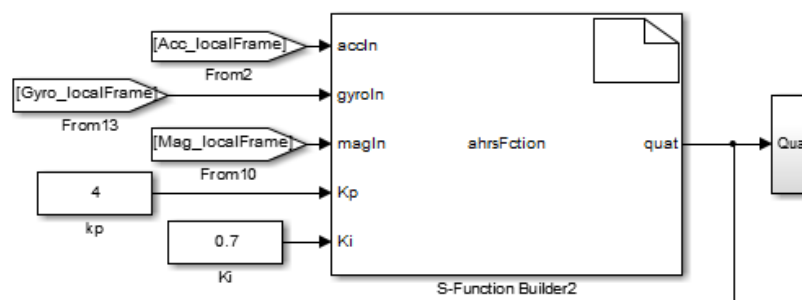
Testy algorytmów orientacji:

Próby odbywały się głównie offline po wcześniejszym zgromadzeniu pomiarów na karcie SD. Wykonywane zostały za pomocą środowiska Matlab Simulink, w którym importowano odczyty z czujników. Próbowano również wykonać tę procedurę online, ale z powodu zbyt dużych opóźnień (słabej jakości komputer na którym uruchomiany był Simulink) i problemów z synchronizacją odczytu po porcie szeregowym porzucono ten pomysł.



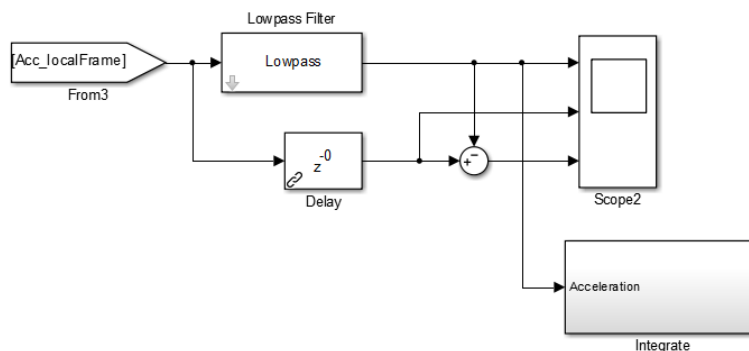
Rysunek 17. Bloki wczytujące, kalibrujące, uśredniające pomiary. Źródło: opracowanie własne.

Testowane algorytmy zostały zaimportowane z języka C w postaci S-funkcji. Zostały one zmodyfikowane tak, by można podawać parametry np. wzmacnienia, czy czasu całkowania bez ponownej kompilacji. W trakcie używania S-funkcji należy zwrócić uwagę, by nie zmieniały one bezpośrednio argumentów wejściowych – Matlab operuje na jednej kopii, więc może to prowadzić do trudnych do zidentyfikowania błędów.

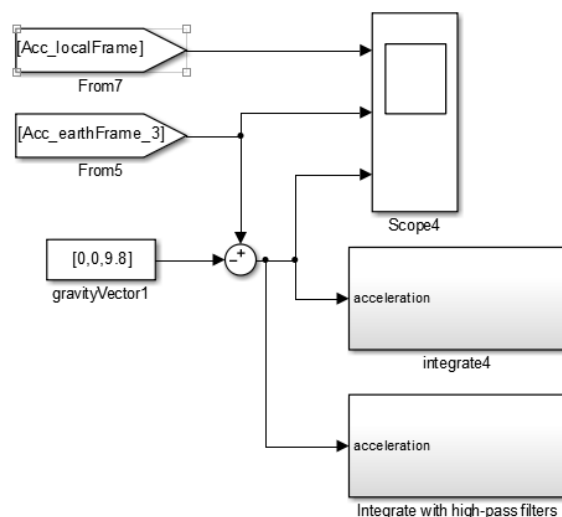


Rysunek 18. S-Funkcja zaimplementowanego w C filtru. Źródło: (opracowanie własne).

Wykonane zostały próby estymacji prędkości i położenia za pomocą odczytu z akcelerometru. Wykonano różne podejścia do tego problemu.



Rysunek 19. Projektowanie filtru do wycięcia składowej stałej z przyspieszenia (grawitacji). Źródło: (opracowanie własne).

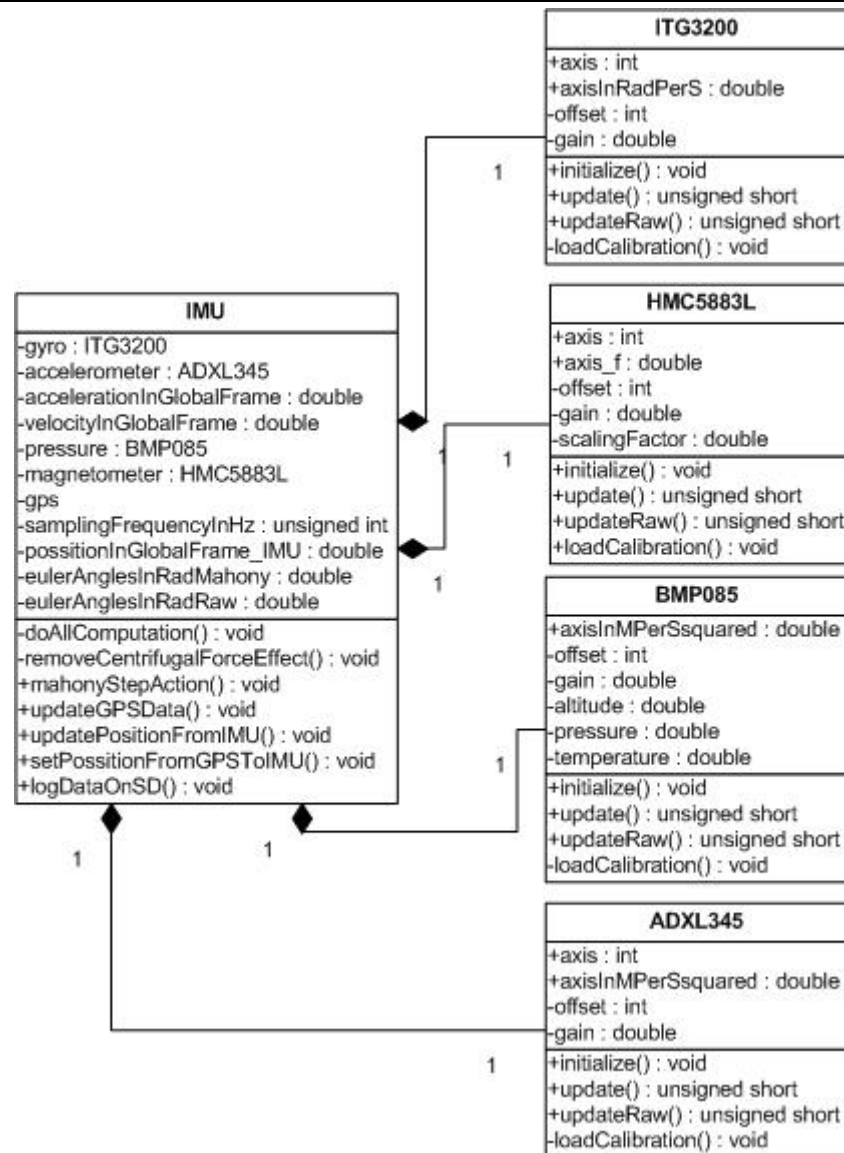


Rysunek 20. Całkowanie przyspieszenia przekształconego do lokalnego ukl. wsp. przy pomocy wartosci z fuzji czujników. Źródło: (opracowanie własne).

Klasy występujące w programie na płycie STM32

Poniżej zostały pokrótce zaprezentowane głównie wykorzystywane klasy w głównym programie. Przedstawiono ich opis, oraz podstawowy diagram UML z kluczowymi metodami i polami składowymi.

- IMU,
 - Klasa łącząca wszystkie elementy – czujniki, komunikacje, filtrację.
 - Zawiera podklasy dla każdego z czujników i filtrów.
 - Jest zmienną globalną typu volatile, gdyż jest wykorzystywana w procedurach przerwań.
 - Do kluczowych operacji zastosowano również podstawowe procedury dostępu do zasobów dzielonych typu mutex.
- Dla każdego z czujników: ADXL345, BMP085, HMC5883L, ITG3200,
 - Zawiera metody do komunikacji, konfiguracji, aktualizacji wartości pomiarów,
 - Posiada algorytmy które wstępnie przetwarzają wartości odczytane z rejestrów na jednostki fizyczne układu SI.



Rysunek 21. Diagram głównej klasy programu - IMU. Źródło: (opracowanie własne)

- Kalman,
 - Zawiera procedury filtrujące do estymacji kątów Eulera na podstawie akcelerometru i żyroskopu.
- Delay,
 - Wykonywanie opóźnień bez obciążania dodatkowego procesora,
 - Współpracuje z mechanizmem przerwania SysTick o znanej częstotliwości.
- GPIO
 - Obsługa na przerwaniach przycisków i diod.
- SdCardLogger
 - Komunikacja po protokole SPI z kartą SD, otwieranie plików, logowanie danych w odpowiednim formacie, nakładka dla metod z biblioteki FatFS.
- SpeedTester
 - Odpowiedzialna za testowanie czasu wykonania poszczególnych elementów implementacji. Oparta na zegarze.

Dodatkowo część bibliotek została napisana w języku C, w stylu klas opartych na plikach nagłówkowych (header .h) i źródłowych (source .c). Część też stanowią biblioteki zapożyczone z innych projektów czy prac naukowych Są to m.in:

- Filtry: mahonyAHRS,
- Obsługa interfejsów I2C, SPI, USART, USB Device Library. W tej ostatniej wprowadzono dość duże modyfikacje i dodatkowe funkcje umożliwiające obsługę ramek w trybie przerwań, oraz wysyłanie w tle, bez blokowania działania programu.

Ramki komunikacyjne:

Do komunikacji pomiędzy komputerem, a układem pomiarowym oraz dodatkowymi elementami systemu zaprojektowano specjalny format ramki. W celu umożliwienia intuicyjnej pracy z konsolą zbudowane są one z znaków ASCII. W planach jest zmiana na format binarny, co zmniejszy rozmiar przesyłanych jednorazowo informacji, lecz wiąże się to z zmniejszeniem szybkości szukania ewentualnych błędów oraz z koniecznością zwrócenia uwagi na takie rzeczy jak do której strony są wyrównane dane (ang. padding), w jakiej kolejności wysyłane są bity (ang. endiannes).

Przykładowa ramka „\$ABG12*”. Format ramki i możliwe komunikaty z ich opisem przedstawiają poniższe tabele:

Tabela 7. Format ramki komunikacyjnej

Element:	Nagłówek startu ['\$']	Adres odbiorcy	Adres nadawcy	Kod komunikatu	Treść komunikatu	Koniec ramki ['*']
Rozmiar:	1 znak	1 znak	1 znak	1 znak	Zależny od kodu	1 znak

Tabela 8. Adresy używanych urządzeń

Adres	Urządzenie
P	(Pecet) - Laptop
C	(Cortex) - Płytki STM32M4f

Tabela 9. Przykładowe rodzaje komunikatów w ramach

Kod komunikatu	Nazwa	Rozmiar komunikatu	Opis
'R'	Request	1	Prośba o wysłanie aktualnych wartości. W zależności od komunikatu: R1 – czyste dane z IMU [Raw data]. R2 – przefiltrowane wartości kwaternionów. R3 – przefiltrowane kąty eulera
'G'	Gather	5	Prośba o zebranie i wysłanie danej ilości czystych pomiarów. G600 – Zgromadź i wyślij 600 próbek. Dla 100 Hz jest to 6 sekund pomiarów.
'D'	Data	Zależnie	Zawiera aktualne wartości, format opisany w tabeli poniżej.
G	GPS	Zależnie	Ramki komunikatów z GPS (wszystkie po kolei co 1 s)
E	Error	Zależnie	Komunikat występującego błędu

L	Log	Zależnie	Komunikat informacyjny. Np. elementy do debugowania przebiegu

Treść komunikatu o kodzie 'D' jest w następującym formacie:

[nazwa pola]:[wartość pola],[nazwa pola]:[wartość pola]...

np. „x:1220,y:342,hdop:121,”,

Tabela 10. Przykładowe opisy komunikatu typu 'D'.

Pole	Opis	Jednostka
x,y,z	Nieprzefiltrowane kąty Eulera względem kolejnych osi (ang. raw)	mili rad
x_c,y_c,z_c	Przefiltrowane kąty Eulera względem kolejnych osi (ang. cocked)	mili rad
q1,q2,q3,q4	Wartości z wektora kwaternionu – wartość przefiltrowana orientacji	-
altG	Wysokość z GPS	m
lonG	Długość geograficzna z GPS (wartości dodatnie dla północy N)	deg
latG	Szerokość geograficzna z GPS (wartości dodatnie dla zachodu W)	deg
dop	Aktualna dokładność współrzędnych z GPS	-
hdop	Aktualna dokładność wysokości z GPS	-
altP	Wysokość z ciśnieniomierza	m
px,py,pz	Położenie z układu IMU w globalnym układzie współrzędnych	m
vx,vy,vz	Prędkości z układu IMU w globalnym układzie współrzędnych	m
dx,dy,dz	Przyrosty z układu IMU w globalnym układzie współrzędnych	m

Przerwania

Obsługa algorytmu jest w dużej mierze oparta na przerwaniach. STM32M4F udostępnia kilka grup konfiguracyjnych do ustalenia priorytetów. Wykorzystano NVIC_PRIORITYGROUP_2. Umożliwia ona podzielenie przerw na 4 priorytety główne (preemption priority) oraz w obrębie każdej 4 podpriorytety (subpriority). Przerwania z grup o niższym numerze mogą wyłączać te o niższych. Natomiast w obrębie jednej grupy przerwania o wyższym numerze czekają na zakończenie aktualnie obsługiwanego przerwania.

Tabela 11. Priorytety przerw

	Preemptive 0	Preemptive 1	Preemptive 2	Preemptive 3
0	SysTick	IMU (TIM3_IRQn)	VCOM (OTG_FS_IRQn)	GPS (USART1)
1		GPIO (TIM4_IRQn)	GPIO (EXTI9_5_IRQn)	
2				
3				

Poprawne skonfigurowanie tego elementu okazało się często kluczowym w wprowadzaniu potencjalnie niewielkich modyfikacji. Często występowały problemy znane od strony teoretycznej występujące w systemach czasu rzeczywistego np. zakleszcze się procedur, lub zbyt częste wyłączenie.

Program na komputerze – GUI (graphical user interface)

Program został napisany w C++/CLI w środowisku Visual Studio 2013. Głównym zadaniem było umożliwienie szybkiego wykrywania ewentualnych błędów i kontroli bez konieczności edytowania kodu i wgrywania go ponownie do pamięci procesora.

Wykorzystane elementy implementacyjne:

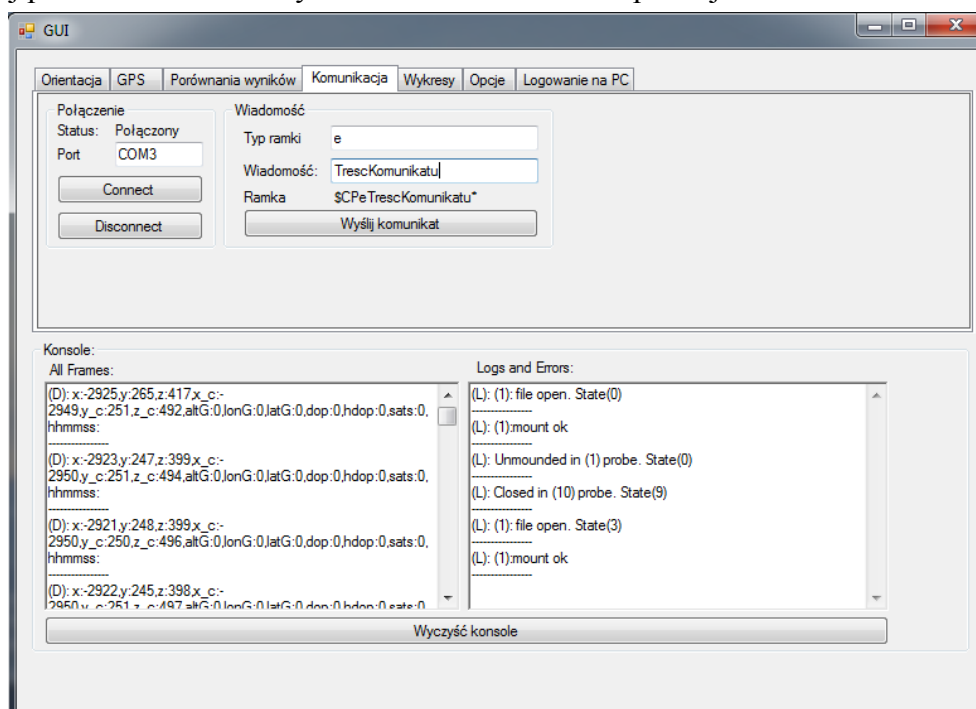
- Obsługa portu szeregowego w trybie przerw,

- Wykorzystanie funkcji delegowanych do aktualizacji wartości w konkretnych elementach GUI. Komunikacja pomiędzy wątkiem „okienka” i przerwań portu szeregowego
- Programowy zegar wywołujący przerwanie z zadaną częstotliwością w którym wysyłane są komunikaty poprzez port szeregowy do płytki STM32.
- Obsługa wyjątków i odpowiednie zaznaczenie ich wystąpienia umożliwiające szybką diagnostykę.

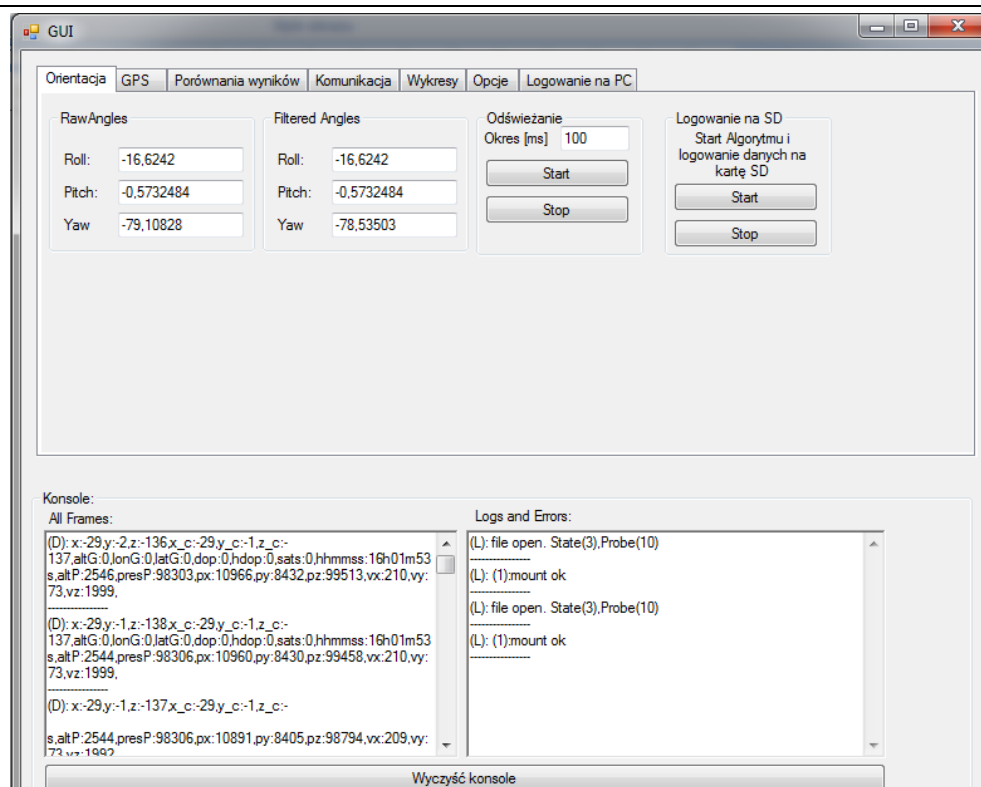
Główne funkcjonalności programu to:

- Połączenie z płytką za pomocą portu szeregowego,
- Wyświetlanie i wizualizacja orientacji – przefiltrowanej i czystych danych,
- Wysyłanie ręczne ramek z poleceniami,
- Logowanie otrzymywanych danych do pliku,
- Dekodowanie ramek,
- Wyświetlanie w konsoli wszystkich otrzymywanych danych,
- Wyświetlanie w osobnej konsoli komunikatów o błędach, przebiegu działania programu. Wybór jakie konkretnie ramki mają być pokazane odbywa się poprzez ich ręczne wpisanie na „twardo” w kodzie programu. Modyfikacja, by można wybrać jakie ramki mają być wyświetlane z poziomu działającej aplikacji jest przewidziana w dalszym rozwoju aplikacji,
- Wybranie szybkości odświeżania wizualizowanych danych – częstotliwość wysyłania zapytań do płytki STM32.

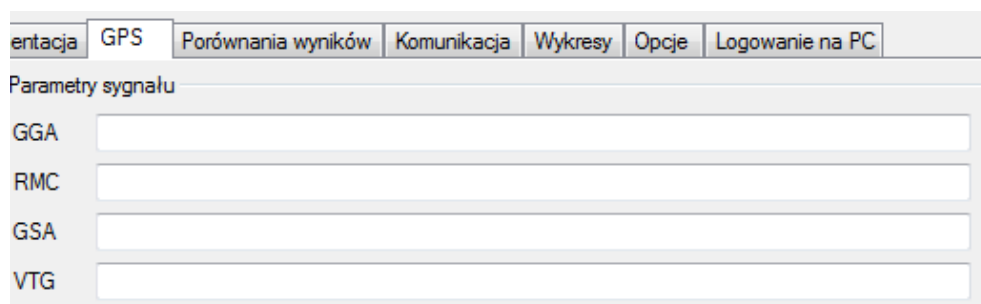
Poniżej przedstawione zostały kilka zrzutów ekranu z aplikacji.



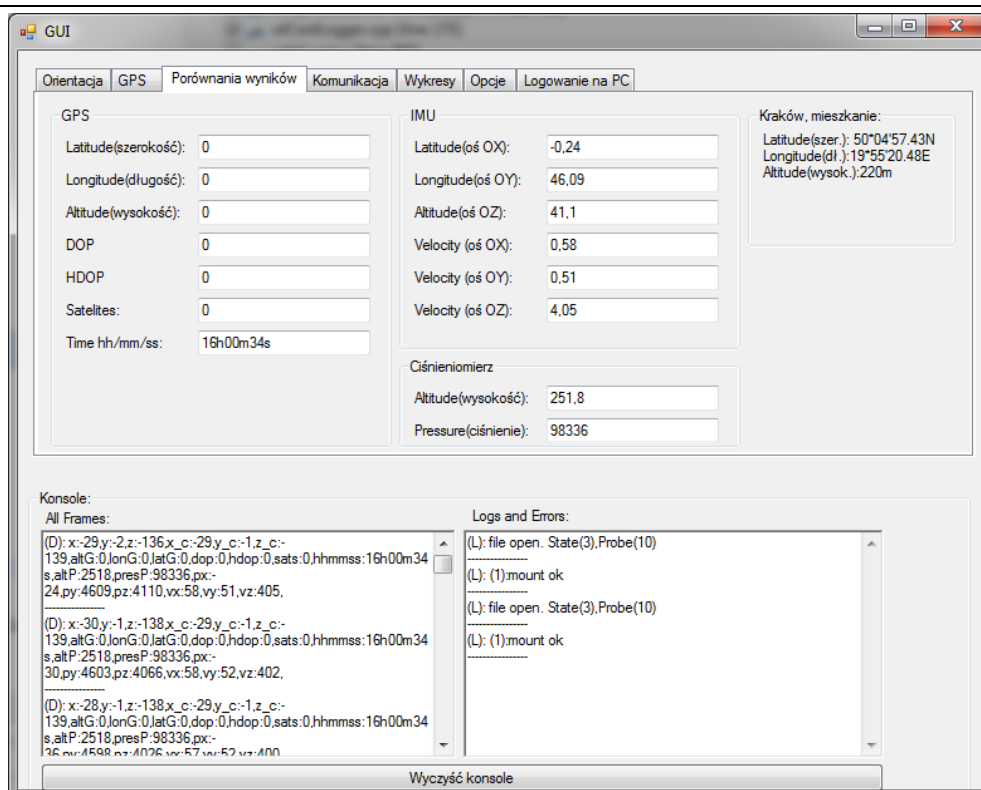
Rysunek 22. Opcja wysyłania komunikatów w poprawnej ramce. Źródło: (opracowanie własne).



Rysunek 23. Odświeżanie i wyświetlanie wartości orientacji Źródło: (opracowanie własne).

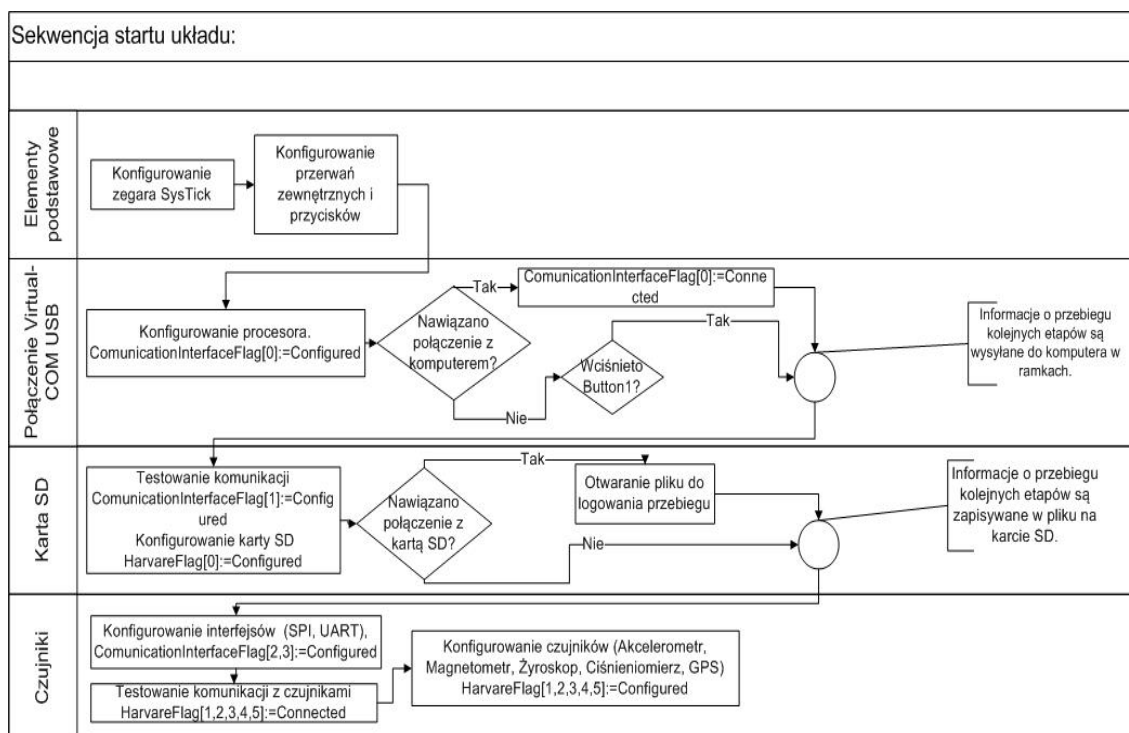


Rysunek 24. Dekodowanie i przedstawienie w wygodnej formie danych GPS. Źródło: (opracowanie własne).



Rysunek 25. Porównanie wartości z GPS'u i układu inercyjnego. Źródło: (opracowanie własne).

5.3 Interakcje i czasowe zależności



Rysunek 26. Sekwencja startu układu. Źródło: (opracowanie własne).

Po zakończeniu inicjalizacji opisanej powyższym diagramem układ przechodzi w stan oczekiwania. Dalsze przebieg zależny jest od komend wysyłanych z programu na komputerze. Gdyby układ nie został podłączony do komputera to poprzez naciśnięcie

przycisku Button1 przejdzie on do pracy standardowej zgodnej z zaprogramowaną aktualnie konfiguracją.

Przez cały etap konfigurowania, komunikacji i pracy właściwej jest dostępna opcja debugowania, czyli zapisywane jest rozpoczęcie i zakończenie działań oraz z jakim skutkiem zostały one zakończone. Odbywa się to na trzy sposoby:

1. Flagi jako zmienne globalne
 - a. Dla kluczowych elementów (np. czujniki, interfejsy komunikacyjne, stan działania algorytmu, stan połączenia z komputerem itd.) stworzone zostały zmienne które ulegają zmianie w kolejnych etapach działania.
2. Logowanie danych do pliku na kartę SD
 - a. Gdy podłączona została karta SD tworzony jest plik gdzie zapisane są wszystkie etapy działania.
 - b. W tym celu stworzono zestaw funkcji, które pozwalają zapisywać te komunikaty w przystępnej formie.
3. Wysyłanie specjalnych ramek („L” i „E”) poprzez USB do programu GUI.
 - a. Gdy wykryto połączenie do programu na komputerze wysyłane mogą być też komunikaty tym kanałem.
 - b. W tym celu stworzono specjalną ramkę, która odbierana w programie wyświetlona zostaje w konsoli.

Możliwe schematy działania po inicjalizacji:

- tryb domyślnej pracy – obliczania orientacji i pozycji, logowanie danych na karcie SD. Uruchamiany komendą, albo przyciskiem gdy brak połączenia do komputera,
- obliczanie orientacji i pozycji i wysyłanie ich z zadaną częstotliwością na komputer w trybie odpytowania (pooling),

Możliwe stany flag w zależności od ich typu:

- Flaga 1(interfejsy): nonInit(0), Configured(1), Error(10)
- Flaga 2(czujniki): nonInit(0), Configured(1), Connected(2), notFound(11), working(5), error(10)
- Flaga 3(podłączenie z komputerem): nonInit(0), Configured(1), ConnectedWithPC(3), ConnectedWithClient(4), Error(10),

Tabela 12. Flagi i przypisane do nich elementy.

Kategoria:	Element:	Zmienna:
Czujniki	SPI1	flagsComunicationInterface[0]
	Akcelerometr	flagsHardware[0]
	Magnetometr	flagsHardware[1]
	Żyroskop	flagsHardware[2]
	Ciśnieniomierz	flagsHardware[3]
GPS	UART1	flagsComunicationInterface[1]
	GPS	flagsHardware[4]
Karta SD	SPI2	flagsComunicationInterface[2]
	Karta SD	flagsHardware[5]

USB – Virtual Com	USB, Połączenie z komputerem	flagsCommunicationInterface[3]
-------------------	------------------------------	--------------------------------

5.4 Podsumowanie rozdziału

Zaprojektowana została cała platforma komunikacyjno-filtrująca, która umożliwia szybkie sprawdzenie w jakim stanie jest każdy z układów. Stworzenie jasnego systemu komunikatów ramek umożliwia szybki dalszy rozwój i monitorowanie oraz sterowanie przebiegiem programu. Lepsza obsługa procedury błędów (inspirowana wyjątkami znanymi z języków wyższego poziomu - wysyłanie informacji z komunikatem) pozwoliła szybko identyfikować pojawiające się problemy. Ilość zastosowanych czujników wymagała zastosowania kilku algorytmów i podejść do ich połączenia. Ogólny podział na klasy pozwolił na lepszą organizację kodu. Zastosowane zostały podstawowe procedury kalibracji, które poprawiają jakość danych.

6 Testy rozwiązania

Część z testów przebiegała tylko podczas tworzenia algorytmów, dlatego tylko niektóre z nich zostały dokładniej opisane. Z powodu braku dostępu do innej platformy o wyższej dokładności, która mogła by służyć jako pomiar referencyjny, doświadczenia są analizowane tylko w sposób ręczny. Każde z wykonanych zadań posiada krótki opis oraz na samym końcu zaznaczono, czy w doświadczeniu wykorzystano tylko układy inercyjne, czy również nawigację satelitarną. Niektóre z nich posiadają również odwołania do innych fragmentów implementacji, gdzie dokładniej zostały one wykorzystane.

6.1 Opis doświadczeń

Testowanie estymacji orientacji:

1. Ruch góra-dół wzdłuż jednej z osi, bez obrotów. (tylko IMU)
Powód wykonania: Dobranie tak współczynników filtrów, by orientacja się nie zmieniała przy tego typu ruchach. Dodatkowo po obrocie przyspieszenia z akcelerometru do globalnego układu odniesienia i odjęciu grawitacji powinno się otrzymać czyste przyspieszenie liniowe.
2. Powolne i szybkie ruchy czujnikiem, obserwacja w trybie online w programie simulnk na wykresie wyników (tylko IMU)
Powód wykonania: wizualna ocena czy współczynniki filtrów są dobrane poprawnie dla całego zakresu, oraz szybki test, poprawności działania całego systemu.
3. Obroty o wielokrotności 90 stopni wokół jednej z osi, logowanie danych i analiza w trybie offline (tylko IMU)
Powód wykonania: Dobieranie parametrów regulatora, sprawdzenie zachowania innych sposobów filtracji, dobranie współczynników wzmocnienia dla żyroskopu (kalibracja).

Testowanie estymacji położenia:

1. Przesunięcie o zadaną odległość w jednej z osi, logowanie danych i analiza w trybie offline (tylko IMU),
 - a. Ruch 20cm na każdej z osi,
 - b. Ruch na większą odległość: np. 50cm stop, -50cm
 - c. Ruch w dwóch osiach – np. po przekątnej prostokąta,
 - d. Ręczne wprowadzenie startowej orientacji – brak błędu związanego z początkowym stabilizowaniem,
2. Dowolny ruch i powrót do pozycji początkowej, analiza w trybie online z opcją zerowania stanu w dowolnym momencie i offline poprzez logowanie danych. Obliczenie błędu i sprawdzenie w jakim momencie nastąpiła największe pogorszenie stanu. (tylko IMU),
3. Przejazd/przejście/przelot na określonych odcinkach i obserwacja ewentualnych anomalii w trybie online oraz opisanie etapów i analiza offline (IMU i GPS),

6.2 Przebieg i wyniki pośrednie

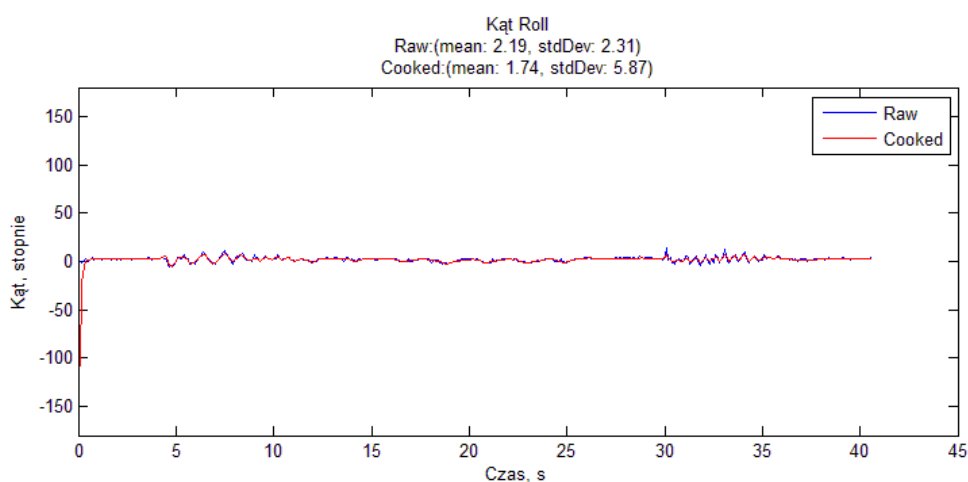
Testowanie algorytmów estymacji orientacji.

1. Doświadczenie typu 1 – ruch oscylacyjny góra-dół. Przykładowy przebieg:

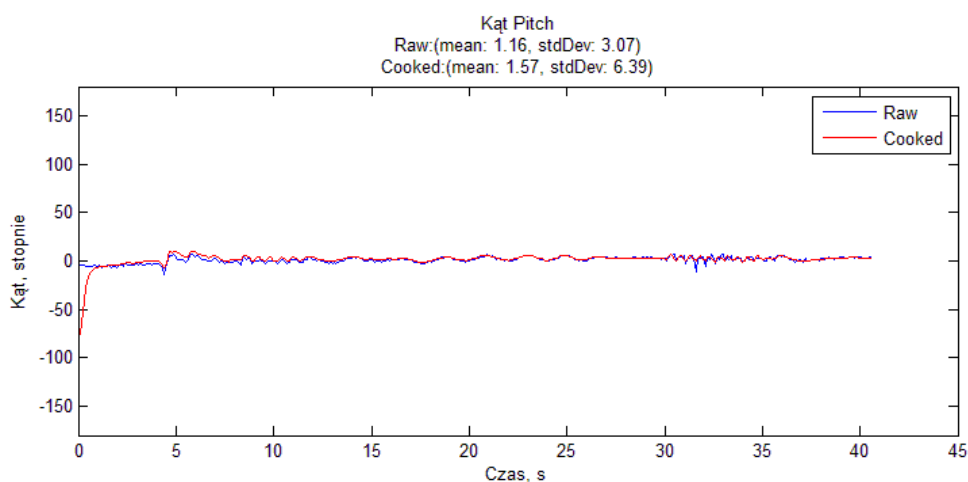
Tabela 13. Przebieg przykładowego doświadczenia estymacji orientacji typu 1.

Czas, s	Opis wykonywanej akcji:
0 – 4	nieruchomy
4 - 9	dość szybkie ruchy góra-dół
9 - 18	nieruchomy
18 - 25	powolne oscylacje góra-dół
25 - 30	nieruchomy
30 - 35	bardzo szybkie oscylacje góra-dół
35 - 40	nieruchomy

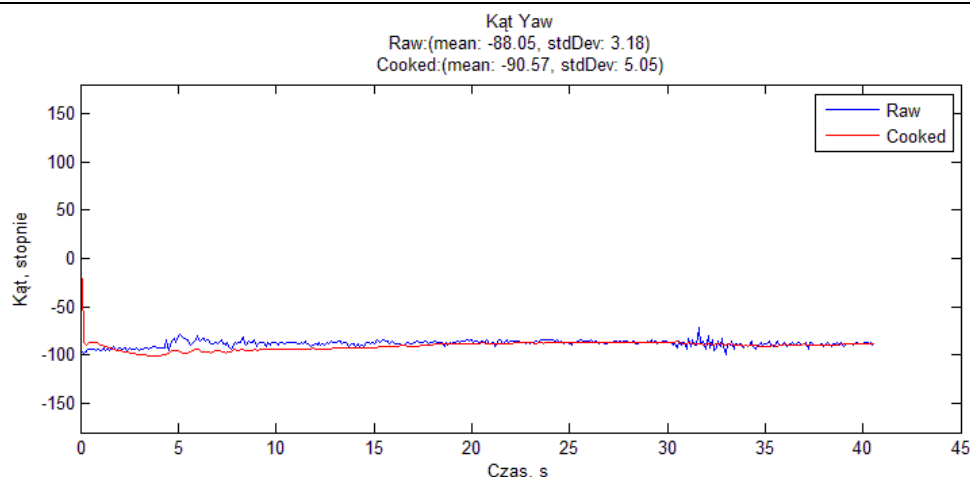
Na wykresach czerwonym kolorem przedstawiono przebieg wartości przefiltrowanej (Cooked), natomiast niebieską wartości tylko oparte na akcelerometrze i magnetometrze (Raw).



Rysunek 27. Zmiana wartości kąta roll w czasie doświadczenia.



Rysunek 28. Zmiana wartości kąta pitch w czasie doświadczenia.



Rysunek 29. Zmiana wartości kąta yaw w czasie doświadczenia.

Zaraz po starcie algorytmu układ musi pozostać chwilę nieruchomo, ponieważ wartości początkowe są inne od aktualnej orientacji, dlatego właściwe doświadczenie musi mieć miejsce po kilku sekundach od startu filtracji.

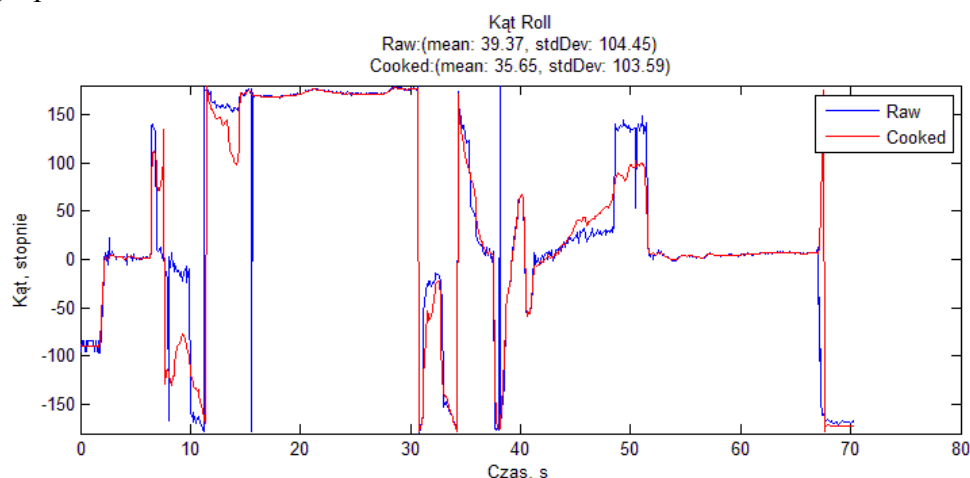
Zauważono, że filtracja kąta yaw jest nieczuła na oscylacje – jest to związane, że głównym źródłem wartości jest odczyt z magnetometru (w takim ułożeniu czujnika i przy takich oscylacjach). Natomiast kąty roll i pitch już ich wartości nie pozostają stałe.

Przy stosunkowo powolnych oscylacjach orientacja ulegała zmianie, lecz już przy bardzo szybkich wartości pozostały stałe, co dobrze świadczy o algorytmie. Potencjalnie będzie on wykluczał chwilowe „turbulencje” z zmieniania wartości estymowanej.

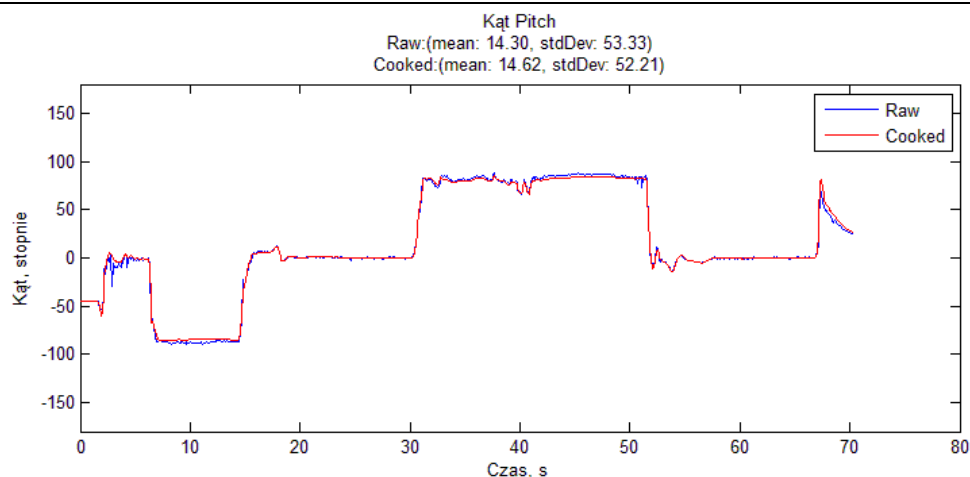
Analogiczne doświadczenia przeprowadzono dla oscylacji w kolejnych osiach.

Ponieważ nie zaimplementowany możliwości zmiany wartości parametrów filtrów (wzmocnienie, czas zdwojenia) w trakcie działania platformy, to przeprowadzenie kolejny raz tego samego doświadczenia wymagało zmiany w kodzie i ponownej kompilacji programu –co znacznie utrudniało prace. W dalszych etapach zaplanowana jest wprowadzenie modyfikacji tego elementu z poziomu GUI.

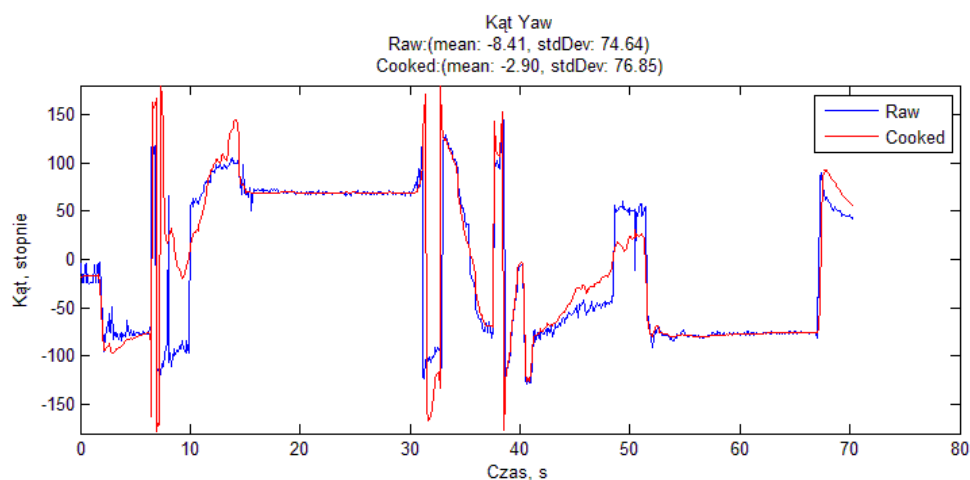
2. Doświadczenie typu 3. Przykładowy przebieg przy obrocie wokół osi OY – zmianie kąta pitch.



Rysunek 30. Zmiana wartości kąta roll w czasie doświadczenia.



Rysunek 31. Zmiana wartości kąta pitch w czasie doświadczenia.



Rysunek 32. Zmiana wartości kąta yaw w czasie doświadczenia.

Testowanie algorytmów estymacji położenia:

1. Doświadczenie typu 1.

Przesunięcie o 20 cm w jednej z osi i następne obliczanie przyspieszenia liniowego i całkowanie w celu otrzymanie aktualnego stanu systemu.

2. Doświadczenie typu 3.

Przejsięcie równolegle do boiska sportowego tam i z powrotem w parku krowoderskim z zatrzymaniem w punkcie startowym i w punkcie zawracania.

Wykonane zostały dwa doświadczenia. Ich przebieg czasowy opisany został poniższymi tabelami.

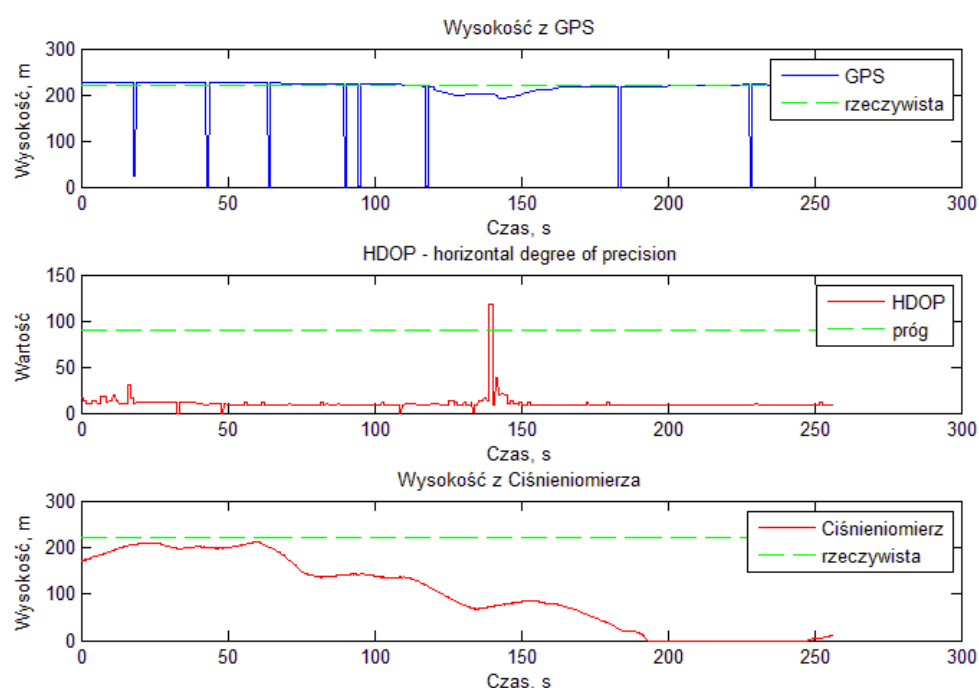
Tabela 14. Przebieg doświadczenia pierwszego estymacji położenia typu 3.

Czas, s	Opis wykonywanej akcji:
Przed rozpoczęciem	Oczekiwanie na złapanie "fixa" i dobrą widoczność satelitów (6-7)
0 – 120	Przejsięcie w jedną stronę
120 – 170	Zatrzymanie się i powolny obrót o 180 stopni
170 – 240	Przejsięcie do miejsca startu

240 – koniec	Zatrzymanie się
--------------	-----------------

Tabela 15. Przebieg doświadczenia drugiego estymacji położenia typu 3

Czas, s	Opis wykonywanej akcji:
Przed rozpoczęciem	Oczekiwanie na złapanie "fixa" i dobrą widoczność satelitów (>7 satelitów)
15 – 95	Przejsię w jedną stronę
95 – 130	Zatrzymanie się i powolny obrót o 180 stopni
130 – 200	Przejsię do miejsca startu
200 – 210	Zatrzymanie się
210	Odłożenie czujników i zatrzymanie logowania



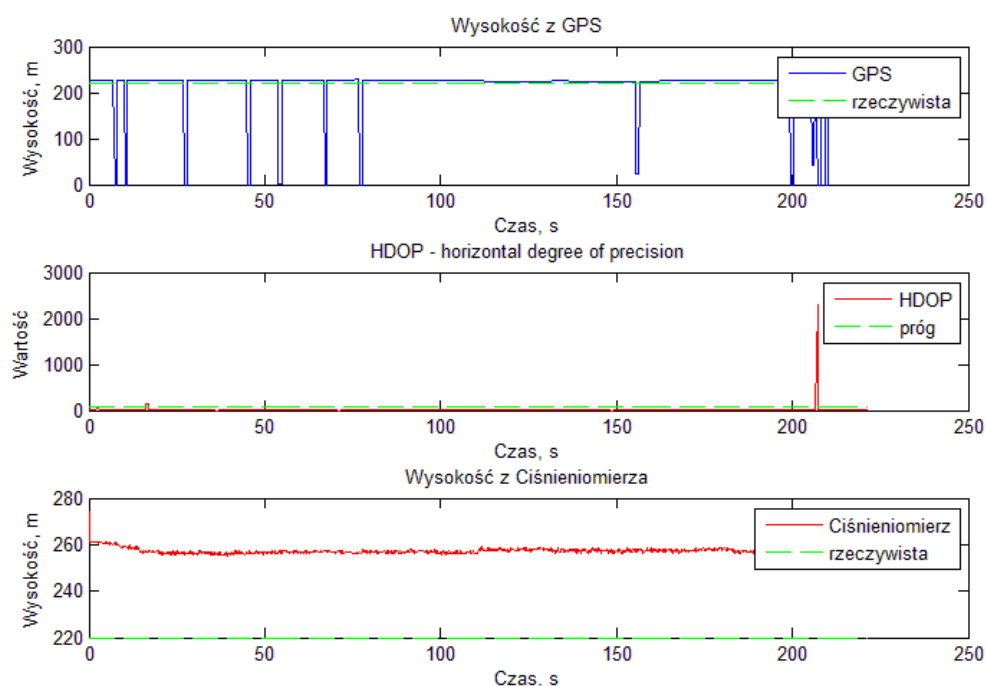
Rysunek 33. Zmierzona wysokość w czasie doświadczenia pierwszego.

W czasie doświadczenia widać, że w momencie, gdy układ pozostawał w ruchu dokładność (HDOP) była na satysfakcjonującym poziomie, natomiast gdy zatrzymano się wzrosła ona powyżej progu jaki przyjęto za graniczny do uznania że pomiary są wartościowe. Potwierdzają to wartości wysokości jaka w tym momencie ulega zmianie.

Zaobserwowano dziwne anomalie w pomiarze ciśnienia. Po pewnym czasie mierzone wartości ciśnienia stawały się zbyt małe i obliczając wysokość wartość stawała się ujemna. Zastanawiając się jakie czynniki mogły spowodować takie zachowanie jako jeden z powodów była dość wysoka temperatura i nasłonecznienie samej płytki a tym samym niestandardowe jej nagrzewanie. Zbadano sam czujnik i okazał się on wrażliwy na zmiany temperatury.

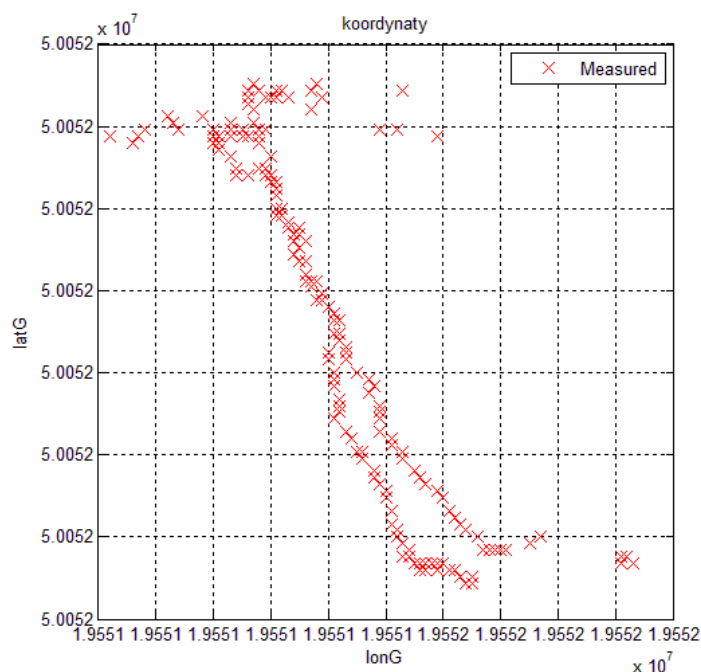
Rozwiązaniem tego problemu okazało się co jakiś czas wykonywanie żądania pomiaru temperatury. Układ ma w sobie wbudowane automatyczne dopasowywanie ciśnienia do aktualnej temperatury na podstawie wartości jaką posiada w danym rejestrze.

Doświadczenie zostało wykonane ponownie (doświadczenie drugie – przebieg w tabeli poniżej) z wprowadzoną modyfikacją w kodzie. Wyniki prezentują się poniżej. Jak widać są wysokość z ciśnieniomierza jest już stabilna. Dodatkowym zagadnieniem jest wartość ciśnienia na wysokości 0m nad poziomem morza. Wzór zaprezentowany w dokumentacji technicznej posiada szacunkową wartość, która nie zawsze jest odpowiednia (inna wilgotność powietrza itd.). Wymagane będzie zaimplementowania procedury kalibracji tej wartości przy starcie z znanej wysokości.

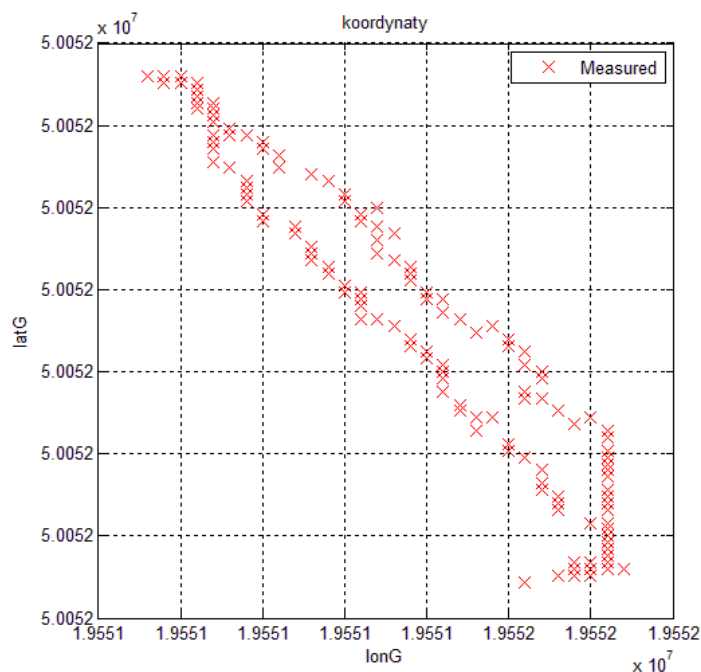


Rysunek 34. Wyniki doświadczenia drugiego, po wprowadzeniu poprawki w kodzie na temperaturę czujnika.

Dane pomiarowe z GPS'u potwierdziły tylko posiadana już wiedzę. Przy pełnej widoczności (6-7 „połączonych,, satelitów) wartości w czasie ruchu było dość dokładne, natomiast w przypadku zatrzymania (krańcowe miejsca na powyższym rysunku) wystąpił rozrzut od rzeczywistej pozycji.

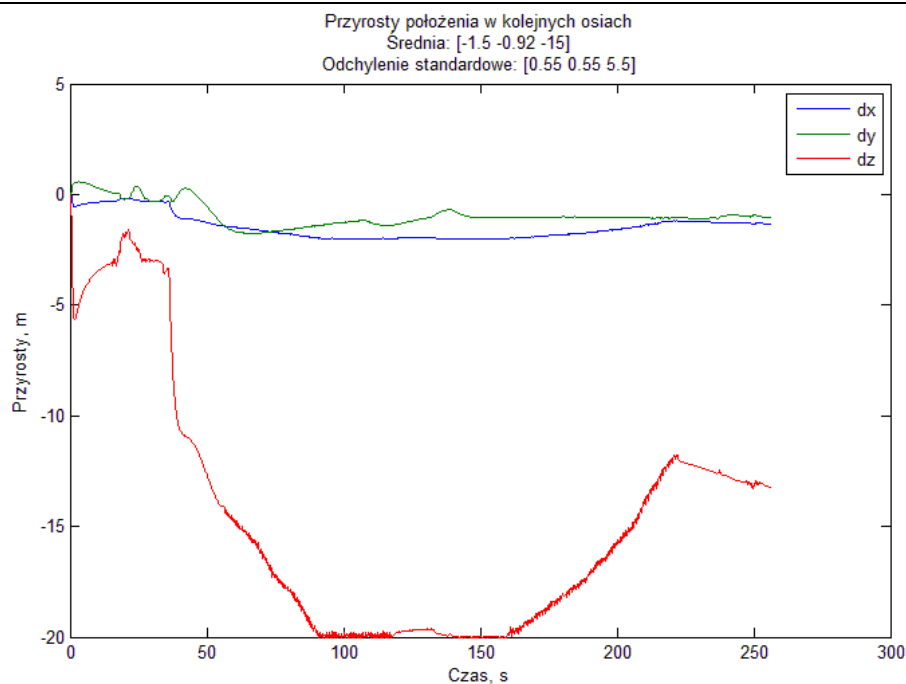


Rysunek 35. Koordynaty z GPS'u w czasie doświadczenia pierwszego.



Rysunek 36. Koordynaty z GPS'u w czasie doświadczenia drugiego.

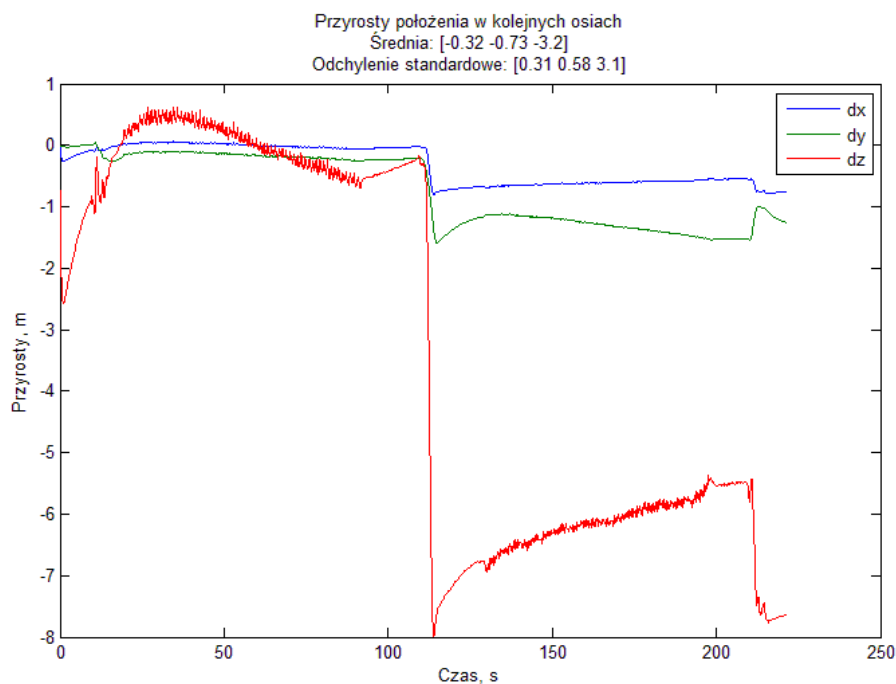
Bardzo niedokładne okazało się podejście z estymacją położenia poprzez całkowanie przyspieszenia (tylko na podstawie czujników MARG). Wartości bardzo szybko osiągały całkiem nierealne wartości co sprawiało, że podejście w tej postaci jest bezwartościowe w dłuższej perspektywie.



Rysunek 37. Przyrosty położenia na osiach – tylko wartości IMU w czasie doświadczenia pierwszego.

Zdecydowanie złe wyniki widać na osi OZ – jest to spowodowane niedokładną kalibracją a tym samym przy kompensacji wartości przyspieszenia grawitacyjnego (odjęcie 9.82) wynik był zawsze ujemny - co cały czas zwiększało błąd.

Mimo wszystko zachowana została pewne charakterystyczne elementy i miejsca zmiany kierunku ruchu. Widać miejsca, gdzie układ się przemieszczał (oscylacje na czerwonym wykresie poniżej 20-100s i 130-200s). Świadczy to o możliwości wykorzystania tego typu pomiaru w przypadku posiadania zdecydowanie wyższej klasy czujników.



Rysunek 38. Przyrosty położenia na osiach – tylko wartości IMU w czasie doświadczenia drugiego.

6.3 Podsumowanie końcowe

Brak dostępu do dokładnej nawigacji GPS oraz czujników IMU spowodował, że doświadczeń nie można obiektywnie ocenić. Próba ręcznego pomiaru i zaplanowania trajektorii ruchu pozwala stworzyć imitację takiego pomiaru referencyjnego, lecz obarczony jest on błędem dokładności późniejszego przebiegu pomiaru. Mimo tego udało się zbadać od praktycznej strony ten typu systemu.

7 Podsumowanie

Pomimo, że projekt nie jest na etapie finalnego produktu, stanowi on dla twórcy dość duży powód do satysfakcji. Pozwolił on zdobyć doświadczenie w zmaganiu się z niestandardowymi problemami, głównie związanymi z szukaniem błędów w aplikacji pracującej na przerwaniach. Dużym wyzwaniem stało się zaprojektowanie i podział na elementy. Pomimo dużego nakładu i chęci zrobienia to w sposób jak najbardziej ogólny i przejrzysty to dopiero po zrobieniu kilku elementów widać, że inne podejście mogło być zdecydowanie lepsze.

Za najbardziej rozwojowy element autor uważa zapoznanie się z bazą artykułów naukowych. W trakcie poszukiwań podobnych rozwiązań i podejść udało się poznać sposoby to robić i czym się w tym czasie kierować. Głównie wykorzystywana została baza IEEE udostępniana przez uczelnię AGH darmowo dla studentów.

Kierunkiem jaki będzie teraz podjęty jest zrobienie odpowiedniej płytki, która będzie łączyła wszystkie elementy, dodanie do niej modułu Bluetooth lub Wifi w celu bezprzewodowej komunikacji. Kolejnym krokiem będzie re-faktoryzacja kodu, ponieważ jest ona stosunkowo niekonsekwentnie stworzona z powodu zmian jakie stały się niezbędne w procesie dodawanie nowych funkcjonalności.

Poniżej zamieszczony został bardziej szczegółowy opis wcześniej wspomnianych problemów i planów na przyszłość dotyczących omówionej w tej pracy platformy.

7.1 Problematyczne elementy

- Słaba jakość czujników, duże błędy i zaszumienie pomiarów.
- Problemy z obsługą karty SD przez interfejs SPI. Pomimo dobrej dokumentacji i wielu przykładów stanowiła ona duży problem. Po wielu godzinach inwestycji problemu okazało się, że biblioteki oferowane przez STM, nie działają w sposób opisany w dokumentacji, co powodowało oczekiwanie na kolejne bajty, albo wysyłanie nadmiarowych informacji, co w sposób losowy wywoływało błędy komunikacji. W końcowym rozrachunku element, który nie był od początku podejrzewany jako niepoprawny okazał się błędny.
- Zakleszczanie się programu w przerwaniach, gdy wykonywane w nim były nieprzemyślane i długie elementy, które często oczekiwały wykonanie innego przerwania o niższym priorytecie.
- Z perspektywy czasu brak odpowiednio dobrego zaplanowania klas i ich relacji powodował wydłużanie czasu implementacji, ale też często zdarzało się, że elementy musiały podlegać gruntownej re-faktoryzacji. Wydłużało to znacznie czas tworzenia nowych funkcjonalności.

7.2 Możliwe drogi rozwoju projektu

- Własne zaprojektowanie płytki z czujnikami obsługującą przerwania generowane przez sensory. Powinno to znacznie zmniejszyć ilość wysyłanych zapytań i ograniczyć komunikację.
- Połączenie układu z modułem GPRS i umożliwienie obserwacji stanu i komunikacji poprzez stację naziemną.
- Zaimplementowanie systemu czasu rzeczywistego (np. freeRTOS), który powinien w znaczny sposób ułatwić wykonywanie równoległe zadań. Teraz dbanie o zapewnienie dostępu do przestrzeni współdzielonych i równoległe wykonywanie zadań musiało być zapewnione przez programistę w sposób jawny. W systemie RTOS te elementy odbywają się praktycznie bez naszej wiedzy.
- Dokładniejsza kalibracja – z uwzględnieniem zmian przesunięcia wartości z temperaturą oraz nieliniowości pomiarów.
- Przejście z ramek w trybie tekstowym ASCII na tryb binarny z wykorzystaniem masek w celu ograniczenia transferu danych. W przyszłości przesyłanie obiektów z komunikatami przy pomocy ich serializacji/deserializacji.
- Wprowadzenie algorytmów autopilota.
- Zakup czujników wyższej precyzji.

8 Cytowane prace

1. **DFRobot.** *10 Dof sensor Rev 1.0 - connections.* 2012.
2. **STM.** *Discovery kit for stm32f407/417 line - data brief.* 2013.
3. **Kolecki, J.** *Small and Low-Cost Mobile Mapping Systems.* 2011.
4. Przegląd i zastosowanie systemów nawigacji satelitarnej. [Online] <http://e-max.net16.net/nawigacja/>.
5. **Eissfeller, Bernd.** *Performance of GPS, GLONASS and Galileo.* Munchen : brak nazwiska, 2007. strony 185-199.
6. **Youssef, Chraïbi.** *Localization in Wireless Sensor Networks.* 2005.
7. **M.I.T.** The Cricket Indoor Location System. [Online] <http://cricket.csail.mit.edu/>.
8. **S., Shuo, i inni.** *Comparison of three Kalman filters for an indoor passive tracking system.* Chicago, IL : brak nazwiska, 2007. strony 284 - 289.
9. Inertial Navigation Systems Information. [Online] http://www.globalspec.com/learnmore/sensors_transducers_detectors/tilt_sensing/inertial_gyros.
10. **Skaloud J., Vallet J., Keller K., Veyssiere G., Kolbl O.** *HELIMAP: Rapid Large Scale Mapping Using Handheld LiDAR/CCD/GPS/INS Sensors on Helicopters.* Long Beach (USA) : ION GNSS Conference, 2005.
11. **Jekeli, C.** *Inertial Navigation Systems with Geodetic Applications.* 2000.
12. **Haiyang Chao, Calvin Coopmans, Long Di, YangQuan Chen.** *A Comparative Evaluation of Low-Cost IMU's for Unmanned Autonomous Systems.* 2010.
13. **Będkowski, dr hab. inż. Krzysztof.** *Teledetekcja i fotogrametria obszarów leśnych.*
14. **Fu, W. W. Wang and L. C.** *Mirror therapy with an exoskeleton upper-limb robot based on IMU measurement system.* 2011.
15. **J. Eriksson, M. J. Matarić, C. J. Winstein.** *Hands-off assistive robotics for post-stroke arm rehabilitation.* 2005.
16. **C.Cifuentes, A.Braidot, L.Rodríguez, M.Frisoli, A.Santiago, A. Frizera.** *Development of a wearable ZigBee sensor system for upper limb rehabilitation robotics.* Roma, Italy : brak nazwiska, 24-2.06.2012.
17. **T.S.Bruggemann, D.G.Greer, R.A.Walker.** GPS Fault detection with IMU and aircraft dynamics. *IEEE Transactions on aerospace and electronic systems.* 2011, Tom 47, 1.
18. **A.Zul Zafar, D.Hazry.** *A simple approach on implementing IMU sensor fusion in PID controller for stabilizing quadrotor flight control.* brak miejsca : IEEE 7th International Colloquium on Signal Processing and its Applications, 2011.
19. **K.King, N.C.Perkins, H.Churchill, R.McGinnis, R.Doss, R.Hickland.** *Bowling ball dynamics revealed by miniature wireless MEMS inertial measurement unit.* brak miejsca : International Sports Engineering Association, 2010.
20. **O.J.Woodman.** *An introduction to inertial navigation.* Cambridge : brak nazwiska, 2007.
21. **P.A.Brudny.** *Czujniki i napędy dla obiektu autonomicznego w przestrzeni trójwymiarowej.* Wrocław : brak nazwiska, 2013.
22. **Madgwick, Sebastian O.H.** *An efficient orientation filter for inertial and inertialmagnetic sensor arrays.* 2010.
23. **K.Lauszus.** [Online] [Zacytowano: 1 12 2014.] <http://blog.tkjelectronics.dk/2012/09/a-practical-approach-to-kalman-filter-and-how-to-implement-it/>.

24. **R.Bieda, R.Grygiel.** *Wyznaczanie orientacji obiektu w przestrzeni z wykorzystaniem naiwnego filtru Kalmana.* Gliwice : brak nazwiska, 2014.
25. **Heinrich, Yannick.** Repozytorium z biblioteką do przeliczania koordynatów na terenie Francji. [Online] <https://github.com/yageek/lambert>.
26. **W.Premierlani.** *Fast Rotations.* 2011.
27. **AKTAT, Ozan.** *ellipsoid_fit2magnetic_data.*
28. **Gawędzki, W.** *Pomiary elektryczne wielkości nieelektrycznych.* 2011.
29. **W.Maziarz, dr inż.** *Sensory w systemach wbudowanych - przegląd nowoczesnych czujników.*
30. **Devices, Analog.** *ADXL345 datasheet.*
31. **M.Wnuk.** *Wykład 6 - Sterowniki Robotów.* Politechnika Wrocławska : brak nazwiska.
32. **InvenSense.** *ITG-3200 product specification revision 1.4.* 2010.
33. **Yongyao Cai, Yang Zhao, Xianfeng Ding, James Fennelly.** Magnetometer basics for mobile phone application. *Electronic products.* February 2012.
34. **Honeywell.** *HMC5883L Datasheet rev B.* 2010.
35. Eclipse, GNU ARM. [Online] <http://gnuarmeclipse.livius.net/>.

9 Spis obrazków i tabel

Rysunek 1. Zasada działania GPS, Źródło:[4]	7
Rysunek 2. Zasada działania DGPS, Źródło:[4]	8
Rysunek 3. Konfiguracja stabilizowana IMU, Źródło: [9].....	9
Rysunek 4. Kąty Eulera. Źródło:[20]	13
Rysunek 5. Składanie rotacji. Źródło:[opracowanie własne]	14
Rysunek 6. Kwaternion jako wektor i kąt obrotu wokół niego. Źródło: [opracowanie własne] ...	14
Rysunek 7. Zamknięty układ regulacji zastosowany w systemie MARG.....	16
Rysunek 8. Ułożenie sensorów na płytce pomiarowej.	19
Rysunek 9. Odpowiedź filtru ($K_p = 1$, $K_i = 0$).....	22
Rysunek 10. Odpowiedź filtru ($K_p = 2$, $K_i = 0$).....	22
Rysunek 11. Odpowiedź filtru ($K_p = 5$, $K_i = 0.7$).....	23
Rysunek 12. Akcelerometr pojemnościowy, Źródło: [28]	25
Rysunek 13. Przybliżenie układu MEMS, Źródło:[29]	26
Rysunek 14. Żyroskop mechaniczny, Źródło:[20]	27
Rysunek 15. Wykorzystanie siły Coriolisa w żyroskopach MEMS, Źródło: [31]	27
Rysunek 16. Prototyp systemu. Płytki STM32M3f Discovery (1), układ GPS (2), antena do GPS(3), układ IMU – czujników (4), adapter karty SD i translator napięć (5). Źródło: (opracowanie własne)	31
Rysunek 17. Bloki wczytujące, kalibrujące, uśredniające pomiary. Źródło: opracowanie własne.	33
Rysunek 18. S-Funkcja zaimplementowanego w C filtru . Źródło: (opracowanie własne).	33
Rysunek 19. Projektowanie filtru do wycięcia składowej stałej z przyspieszenia (grawitacji). Źródło: (opracowanie własne).	33
Rysunek 20. Całkowanie przyspieszenia przekształconego do lokalnego ukł. wsp. przy pomocy wartości z fuzji czujników. Źródło: (opracowanie własne).	34

Rysunek 21. Diagram głównej klasy programu - IMU. Źródło: (opracowanie własne).....	35
Rysunek 22. Opcja wysyłania komunikatów w poprawnej ramce. Źródło: (opracowanie własne).	38
Rysunek 23. Odświeżanie i wyświetlanie wartości orientacji Źródło: (opracowanie własne).	39
Rysunek 24. Dekodowanie i przedstawienie w wygodnej formie danych GPS. Źródło: (opracowanie własne).....	39
Rysunek 25. Porównanie wartości z GPS'u i układu inercyjnego. Źródło: (opracowanie własne).	40
Rysunek 26. Sekwencja startu układu. Źródło: (opracowanie własne).....	40
Rysunek 27. Zmiana wartości kąta roll w czasie doświadczenia.....	44
Rysunek 28. Zmiana wartości kąta pitch w czasie doświadczenia.	44
Rysunek 29. Zmiana wartości kąta yaw w czasie doświadczenia.....	45
Rysunek 30. Zmiana wartości kąta roll w czasie doświadczenia.....	45
Rysunek 31. Zmiana wartości kąta pitch w czasie doświadczenia.	46
Rysunek 32. Zmiana wartości kąta yaw w czasie doświadczenia.....	46
Rysunek 33. Zmierzona wysokość w czasie doświadczenia pierwszego.	47
Rysunek 34. Wyniki doświadczenia drugiego, po wprowadzeniu poprawki w kodzie na temperaturę czujnika.	48
Rysunek 35. Koordynaty z GPS'u w czasie doświadczenia pierwszego.....	49
Rysunek 36. Koordynaty z GPS'u w czasie doświadczenia drugiego.	49
Rysunek 37. Przyrosty położenia na osiach – tylko wartości IMU w czasie doświadczenia pierwszego.....	50
Rysunek 38. Przyrosty położenia na osiach – tylko wartości IMU w czasie doświadczenia drugiego.....	50
 Tabela 1. Kategorie układów inercyjnych, Źródło: [11], [12]:	10
Tabela 2. Odległość pomiędzy środkami czujników wzdłuż kolejnych osi:	19
Tabela 3. Główne właściwości akcelerometru ADXL345.....	26
Tabela 4. Główne właściwości żyroskopu ITG3200	28
Tabela 5. Główne właściwości magnetometru HMC5883L	28
Tabela 6. Fizyczne podłączenie elementów do płytki	31
Tabela 7. Format ramki komunikacyjnej	36
Tabela 8. Adresy używanych urządzeń.....	36
Tabela 9. Przykładowe rodzaje komunikatów w ramach.....	36
Tabela 10. Przykładowe opisy komunikatu typu 'D'.....	37
Tabela 11. Priorytety przerwania.....	37
Tabela 12. Flagi i przypisane do nich elementy.....	41
Tabela 13. Przebieg przykładowego doświadczenia estymacji orientacji typu 1.	44
Tabela 14. Przebieg doświadczenia pierwszego estymacji położenia typu 3.	46
Tabela 15. Przebieg doświadczenia drugiego estymacji położenia typu 3	47