

Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Paweł Stawicki

Nr albumu: 189254

**Wyznaczanie pozycji oraz orientacji
w przestrzeni za pomocą
ultradźwięków**

Praca magisterska
na kierunku INFORMATYKA

Praca wykonana pod kierunkiem
dra inż. Marcina Peczarskiego
Instytut Informatyki

Październik 2015

Oświadczenie kierującego pracą

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data

Podpis kierującego pracą

Oświadczenie autora (autorów) pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data

Podpis autora (autorów) pracy

Streszczenie

W pracy przedstawiono prototyp urządzenia potrafiącego określić zarówno swoje położenie, jak i orientację w przestrzeni. Urządzenie składa się z dwóch części: nadajnika, którego pozycja i orientacja jest wyznaczana, oraz odbiornika, który stanowi stały punkt odniesienia. Wyznaczanie położenia odbywa się za pomocą pomiarów odległości pomiędzy nadajnikiem a odbiornikiem – wykorzystuje się do tego ultradźwięki o częstotliwości 40 kHz. Zastosowana metoda umożliwia pomiar z wysoką rozdzielczością dochodzącą do 0,5 mm.

Słowa kluczowe

lokalizacja, położenie, orientacja, ultradźwięki, 3D

Dziedzina pracy (kody wg programu Socrates-Erasmus)

11.3 Informatyka

Klasyfikacja tematyczna

- Information systems
 - Information systems applications
 - Spatial-temporal systems
 - Global positioning systems

Tytuł pracy w języku angielskim

Determining position and orientation in space using ultrasound

Spis treści

Wprowadzenie	5
1. Podstawy teoretyczne	9
1.1. Wyznaczanie położenia na podstawie odległości od punktów stałych	9
1.2. Wyznaczanie orientacji w przestrzeni	10
1.3. Pomiar odległości za pomocą ultradźwięków	11
2. Budowa prototypu	13
2.1. Budowa i zasada działania nadajnika	13
2.2. Dobór rezonatorów piezoelektrycznych	14
2.3. Odbiornik	16
2.4. Budowa i zasada działania odbiornika	16
2.5. Budowa modułu ultradźwiękowego	17
2.6. Przystawka do <i>stm32f4-discovery</i>	18
3. Przetwarzanie danych zebranych z odbiornika	21
3.1. Komunikacja z odbiornikiem, moduł <i>com.py</i>	21
3.2. Wyznaczanie odległości, moduł <i>find_pattern.py</i>	23
3.3. Wyznaczanie pozycji oraz orientacji przestrzennej, moduł <i>xyz.py</i>	25
3.4. Kalibracja	25
3.5. Obsługa programu <i>scan.py</i>	26
4. Podsumowanie	29
A. Załączniki	31
Bibliografia	33

Wprowadzenie

W dzisiejszym świecie wyraźnie wzrasta zapotrzebowanie na urządzenia, które potrafią określić swoje położenie i orientację w przestrzeni. Urządzenia takie mają szerokie zastosowanie w wielu dziedzinach, m.in. w wirtualnej rzeczywistości, rozszerzonej rzeczywistości, kartografii lub podczas trójwymiarowego skanowania obiektu. Przykładowo okulary do wirtualnej rzeczywistości, takie jak *Oculus Rift* [4] czy *castAR* [21], muszą uwzględnić położenie i orientację głowy, aby na tej podstawie wyświetlić użytkownikowi odpowiednią treść.

Powstało wiele rozwiązań tego problemu. Do najczęściej stosowanych możemy zaliczyć:

1. Wykorzystanie akcelerometrów, żyroskopów i magnetometrów – takie rozwiązanie zastosowano w *Oculus Rift development kit*. Zaletami tej metody są stosunkowo prosta konstrukcja i niska cena. Do wad należy zaliczyć brak stałych punktów odniesienia, co skutkuje występowaniem tzw. dryftu. *Oculus Rift development kit* radzi sobie z tym problemem, modelując w komputerze możliwe zmiany pozycji głowy. Jednak rozwiązanie to jest dalekie od idealnego, o czym może świadczyć fakt, że w kolejnej wersji urządzenia dodano zewnętrzną kamerę śledzącą pozycję głowy.
2. Projektowanie światła strukturalnego na otoczenie i zbieranie informacji o strukturze światła odbitego za pomocą sensorów, zazwyczaj kamer – taką metodę wykorzystano w *Microsoft Kinect* [12]. Urządzenie wyświetla na otoczenie stały wzór punktów, następnie kamerą na podczerwień zbiera informację o zniekształceniu danego wzoru i na tej podstawie odtwarza trójwymiarową strukturę otoczenia i położenie urządzenia w tym otoczeniu. Podobną metodę wykorzystuje *Oculus Rift development kit 2* [27] oraz *castAR* [21] – tutaj za źródła światła służą diody podczerwone umieszczone na okularach. Emitowane przez nie światło rejestruje kamera umieszczona przed użytkownikiem. Komputer na podstawie względnego położenia widocznych punktów określa położenie i orientację okularów w przestrzeni. Zaletą tego rozwiązania są stałe punkty odniesienia (kamera), a także możliwość pomiaru wielu punktów naraz. Wadami są stosunkowo niska rozdzielczość, szczególnie w osi Z, i duży strumień danych do obróbki.
3. Wykorzystanie wielu zdjęć, na podstawie których wyznaczana jest pozycja kamery względem znajdujących się na nich stałych (niezmieniających się w czasie) obiektów. Możliwa jest też odwrotna sytuacja, gdy kamera (lub wiele kamer) jest punktem stałym i względem niej wyznaczana jest pozycja fotografowanych obiektów – taką metodę wykorzystano w *VidialSFM* [2] oraz w *The Pi 3D scanner project* [31]. Wadą tego rozwiązania jest dość niska rozdzielczość.
4. Pomiar czasu, jakiego potrzebują fale elektromagnetyczne na dotarcie od nadajników o znanych położeniach do obiektu, którego położenie nas interesuje. Dzięki znajomości czasów oraz prędkości rozchodzenia się fal można wyznaczyć odległości pomiędzy nadajnikami a obiektem. Na podstawie odległości wyznaczana jest lokalizacja przestrzenna obiektu. Tę metodę powszechnie stosuje się w takich systemach jak GPS [24],

GLONASS [25] czy Galileo [23]. Jej zaletą jest możliwość lokalizowania obiektów na dużych przestrzeniach. Jednak ze względu na dużą prędkość, z jaką rozchodzą się fale elektromagnetyczne, metoda ta cechuje się dość niską dokładnością nieprzekraczającą 10 cm.

5. Pomiar siły sygnału radiowego (fal elektromagnetycznych) docierającego do obiektu, którego lokalizacja nas interesuje. Źródłem sygnału mogą być dedykowane radiolatarnie, np. *Estimote Beacon* [22], czy inne nadajniki, np. WiFi [10]. Wadę tego rozwiązania stanowi niska dokładność pomiaru, która wynosi maksymalnie kilkadziesiąt centymetrów.
6. Wykorzystanie światła podczerwonego naturalnie emitowanego przez badany obiekt – takie podejście zaprezentowano w pracy *PIR Sensors: Characterization and Novel Localization Technique* [8]. Ta metoda jest podobna do metody drugiej, z tą różnicą, że zrezygnowano w niej ze sztucznego źródła światła, zamiast niego wykorzystując naturalne światło podczerwone emitowane przez badany obiekt. Tutaj również uzyskuje się stosunkowo niską dokładność pomiarową.
7. Wykorzystanie indukcji magnetycznej – metoda polega na pomiarze siły zmiennego pola magnetycznego niskiej częstotliwości generowanego przez zewnętrzne anteny. Na podstawie siły pola docierającego do badanego obiektu estymowane są jego odległości od anten, a na ich podstawie wyznaczana jest lokalizacja przestrzenna obiektu. Główną zaletę tego podejścia stanowi fakt, że pole magnetyczne przenika przez większość materiałów. Pozwala to na lokalizowanie obiektu również pod ziemią, co zaprezentowano w pracy *Revealing the hidden lives of underground animals using magneto-inductive tracking* [7]. Wadą metody jest mały zasięg, ponieważ pole magnetyczne propaguje się odwrotnie proporcjonalnie do sześciannu odległości od źródła.
8. Wykorzystanie specyficznych warunków zewnętrznych, w jakich może znajdować się obiekt, którego lokalizacja nas interesuje – w pracy *We Can Track You If You Take the Metro: Tracking Metro Riders Using Accelerometers on Smartphones* [5] autorzy opisali algorytm lokalizowania użytkownika metra na podstawie danych z akcelerometru, które zmieniały się podczas jazdy. Metoda ta jest jednak dość specyficzna i nie nadaje się do ogólnych zastosowań.

W niniejszej pracy przedstawiono prototyp, który jest oparty na zmodyfikowanej czwartej metodzie i zamiast fal elektromagnetycznych wykorzystuje ultradźwięki. Podejście to zapewnia dużą dokładność, szczególnie w osi Z, prostotę budowy, a także niską cenę. Urządzenie składa się z dwóch części: odbiornika, na którym umieszczono trzy mikrofony, oraz nadajnika, na którym znajdują się cztery głośniki. Pomiedzy mikrofonami i głośnikami mierzy się odległość z rozdzielczością dochodzącą do 0,5 mm, dzięki czemu prototyp jest w stanie określić położenie nadajnika w przestrzeni i jego orientację. Pomiaru odległości dokonuje się za pomocą ultradźwięków o częstotliwości 40 kHz. Mimo iż ta metoda jest znana od wielu lat, to rzadko się ją stosuje do precyzyjnych pomiarów z uwagi na relatywnie dużą długość fali ultradźwiękowej, która dla częstotliwości 40 kHz wynosi około 8,5 mm.

Autorowi niniejszej pracy udało się to pozorne ograniczenie przewyciężyć. Ostatecznie urządzenie jest w stanie śledzić położenie nadajnika z rozdzielczością $5000 \text{ px} \times 5000 \text{ px} \times 5000 \text{ px}$ w przestrzeni ograniczonej sześciannem o rozmiarach $2,5 \text{ m} \times 2,5 \text{ m} \times 2,5 \text{ m}$, przy czasie odświeżania rzędu 350 ms. Warto podkreślić, że przyjęte parametry sześciannu wynikają ze względów praktycznych i bez problemu można je zwiększyć, zachowując stosunek rozdzielczości na metr. Relatywnie długi czas odświeżania zależy głównie od czasu, jakiego potrzebuje

fala dźwiękowa, by się rozproszyć, tak by jej odbicia od powierzchni ścian nie wpływały na kolejne pomiary.

Należy nadmienić, że wykorzystanie ultradźwięków do orientacji w terenie jest powszechnie znane, z powodzeniem wykorzystują je m.in. nietoperze czy niektóre gatunki ryjówek podczas echolokacji. Stosuje się je również w przemyśle, głównie do pomiaru odległości, np. w samochodach podczas parkowania ultradźwiękami mierzona jest odległość od otaczających je obiektów. Powstało także kilka prac wykorzystujących ultradźwięki do lokalizacji przestrzennej. Przykładowo w pracy *Ultrasonic 3D Wireless Computer Mouse* [1] przedstawiono prototyp myszki 3D, której pozycja w przestrzeni określana jest za pomocą ultradźwięków, niemniej wykorzystany przez autorów algorytm wyznaczania odległości jest dużo bardziej podatny na błędy od metody prezentowanej w niniejszej pracy. Ponadto autorzy koncentrują się jedynie na określaniu położenia myszki w przestrzeni, bez wyznaczania jej orientacji. Kolejne ciekawe rozwiązanie stanowi rękawica dla graczy *Power Glove* [28], która pojawiła się w sprzedaży w 1989 roku [29]. Ona również wykorzystuje ultradźwięki do wyznaczania pozycji w przestrzeni, jednak nie odniosła znaczącego sukcesu komercyjnego ze względu na wyjątkowo niską precyzję urządzenia.

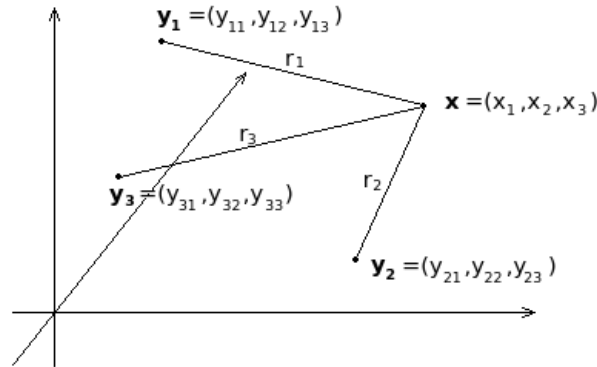
Niniejsza praca podzielona została na cztery rozdziały. W pierwszym rozdziale przedstawiono podstawy teoretyczne lokalizacji oraz orientacji przestrzennej. W podrozdziale 1.3 przedstawiono sposób pomiaru odległości za pomocą ultradźwięków, pokazano również główny czynnik wpływający na dokładność pomiaru. Rozdział drugi zawiera opis budowy oraz zasadę działania prezentowanego prototypu. W rozdziale trzecim przedstawiono zastosowaną metodę przekształcenia zebranych danych ultradźwiękowych na pozycję oraz orientację prototypu. Rozdział czwarty podsumowuje prezentowane podejście, wyliczając jego zalety, jak i wady.

Rozdział 1

Podstawy teoretyczne

1.1. Wyznaczanie położenia na podstawie odległości od punktów stałych

Zadaniem prezentowanego prototypu jest wyznaczanie położenia nadajnika na podstawie jego odległości od znanych punktów stałych. Niech \mathbf{x} będzie szukanim punktem w przestrzeni, $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3$ stałymi punktami o znanym położeniu, a r_1, r_2, r_3 danymi odległościami (rysunek 1.1).



Rysunek 1.1: Wyznaczanie położenia na podstawie odległości od punktów stałych

Wtedy \mathbf{x} możemy wyznaczyć za pomocą układu równań:

$$|\mathbf{y}_1 - \mathbf{x}| = r_1,$$

$$|\mathbf{y}_2 - \mathbf{x}| = r_2,$$

$$|\mathbf{y}_3 - \mathbf{x}| = r_3,$$

co dla przestrzeni euklidesowej daje:

$$(y_{11} - x_1)^2 + (y_{12} - x_2)^2 + (y_{13} - x_3)^2 = r_1^2,$$

$$(y_{21} - x_1)^2 + (y_{22} - x_2)^2 + (y_{23} - x_3)^2 = r_2^2,$$

$$(y_{31} - x_1)^2 + (y_{32} - x_2)^2 + (y_{33} - x_3)^2 = r_3^2.$$

Zauważmy, że znając odległość \mathbf{x} od jednego z punktów stałych, wiemy, że \mathbf{x} będzie leżał na sferze o promieniu równym danej odległości. Znając dwie odległości, możemy ograniczyć poszukiwania do części wspólnej dwóch sfer, czyli zazwyczaj okręgu. Dla znanych trzech odległości dostajemy część wspólną okręgu ze sferą, czyli na ogół dwa możliwe punkty. Mimo iż trzy odległości nie wystarczą do wyznaczenia \mathbf{x} , to otrzymane rozwiązania są na ogół od siebie odległe i możemy założyć, że jedno z rozwiązań jest szukany przez nas punktem.

Ponieważ mamy pełną dowolność w doborze punktów stałych \mathbf{y}_i , dla uproszczenia możemy przyjąć: $\mathbf{y}_1 = (\frac{a}{2}, 0, 0)$, $\mathbf{y}_2 = (-\frac{a}{2}, 0, 0)$, $\mathbf{y}_3 = (0, h, 0)$. Zatem $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3$ są wierzchołkami trójkąta równoramiennego o wysokości h i podstawie a , natomiast układ równań upraszcza się do postaci:

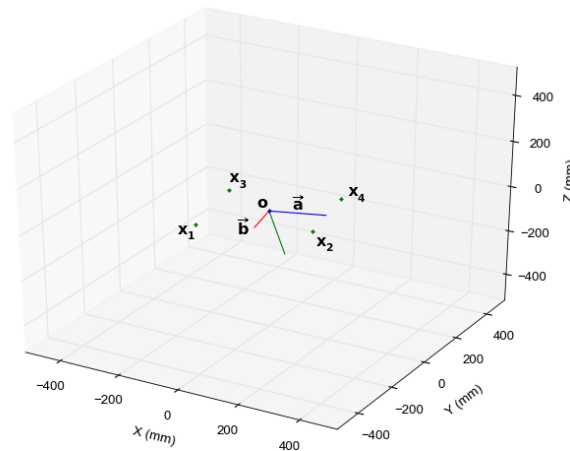
$$\begin{aligned}(\frac{a}{2} - x_1)^2 + x_2^2 + x_3^2 &= r_1^2, \\(\frac{a}{2} + x_1)^2 + x_2^2 + x_3^2 &= r_2^2, \\x_1^2 + (h - x_2)^2 + x_3^2 &= r_3^2,\end{aligned}$$

z czego ostatecznie otrzymujemy:

$$\begin{aligned}x_1 &= \frac{r_2^2 - r_1^2}{2a}, \\x_2 &= \frac{r_2^2 - r_3^2 - ax_1 - \frac{a^2}{4}}{2h} + \frac{h}{2}, \\x_3 &= \pm \sqrt{r_3^2 - x_1^2 - (h - x_2)^2}.\end{aligned}$$

1.2. Wyznaczanie orientacji w przestrzeni

Oprócz położenia nadajnika w przestrzeni interesuje nas również jego orientacja. Standardowo definiuje się ją jako kąty Eulera [26]: przechylenia, pochylenia oraz odchylenia wokół osi związanych z obiektem, jednak na potrzeby niniejszej pracy wygodniej posługiwać się dwoma prostopadłymi wektorami, które jednoznacznie wyznaczają owe kąty. Prezentowany prototyp określa położenie czterech punktów $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$ na nadajniku tworzących prostokąt, a następnie na ich podstawie wyznacza poszukiwane dwa prostopadłe wektory, które jednoznacznie definiują orientację nadajnika (rysunek 1.2).



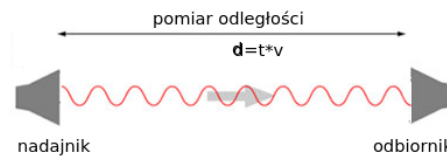
Rysunek 1.2: Wyznaczanie orientacji nadajnika na podstawie czterech punktów

Na rysunku punkt $\mathbf{o} = (\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3 + \mathbf{x}_4)/4$ jest szukanym położeniem nadajnika, a wektory $\vec{\mathbf{a}} = (\mathbf{x}_2 + \mathbf{x}_4)/2 - \mathbf{o}$ oraz $\vec{\mathbf{b}} = (\mathbf{x}_1 + \mathbf{x}_3)/2 - \mathbf{o}$ wyznaczają jednoznacznie orientację przestrzenną nadajnika.

Zauważmy, że wyznaczenie położenia czterech punktów w przestrzeni daje nam pewną nadmiarowość danych. Do wyznaczenia pozycji nadajnika potrzebujemy jedynie trzech współrzędnych, a dla orientacji – zaledwie trzech kątów, czyli w sumie sześciu wartości. Prezentowany prototyp wyznacza dwanaście niewiadomych, co daje dwukrotną nadmiarowość danych. Jest to efekt pożądany, gdyż pozwala na korektę błędów pomiarowych.

1.3. Pomiar odległości za pomocą ultradźwięków

Prezentowane urządzenie do pomiaru odległości wykorzystuje metodę pomiaru czasu, jakiego potrzebuje dźwięk, aby pokonać drogę od nadajnika do odbiornika (rysunek 1.3).



Rysunek 1.3: Pomiar odległości za pomocą ultradźwięków

Znając prędkość rozchodzenia się fal dźwiękowych w powietrzu oraz czas, jakiego fala dźwiękowa potrzebowała do pokonania danego dystansu, jesteśmy w stanie wyznaczyć szukaną odległość.

Prędkość dźwięku w powietrzu w dużym stopniu zależy od panujących warunków atmosferycznych [3], zwłaszcza temperatury. Dla gazu doskonałego prędkość ta wyraża się wzorem:

$$V = 331,3 \sqrt{1 + \frac{T}{273,15}} \quad \left[\frac{\text{m}}{\text{s}} \right],$$

gdzie T jest temperaturą w stopniach Celsjusza ($^{\circ}\text{C}$). Wzór ten możemy przybliżyć za pomocą rozwinięcia Taylora dla temperatur bliskich temperaturze pokojowej 25°C :

$$V = 346,13 + 0,580(T - 25^{\circ}) \quad \left[\frac{\text{m}}{\text{s}} \right]$$

Mimo że współczynnik temperaturowy jest stosunkowo mały i stanowi jedynie 0,17% całej prędkości, to przy odległościach rzędu metrów i rozdzielczościach rzędu milimetrów staje się on istotny, dlatego w przypadku prezentowanego prototypu niezbędna jest wstępna kalibracja, która pozwala wyznaczyć aktualną prędkość dźwięku.

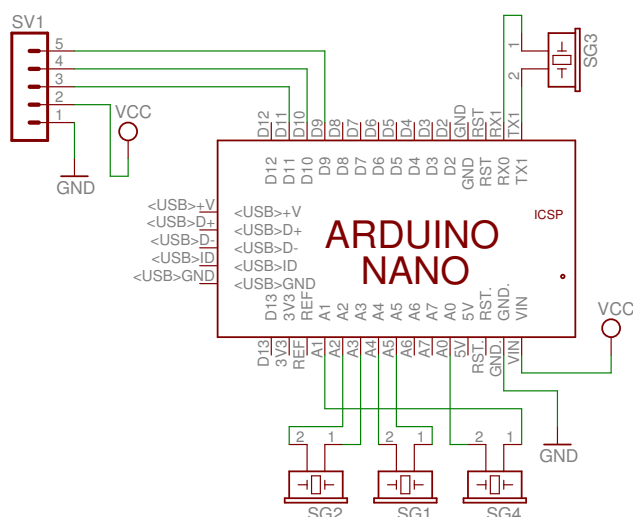
Rozdział 2

Budowa prototypu

Prezentowany prototyp składa się z dwóch części: nadajnika i odbiornika, pomiędzy którymi dokonywany jest pomiar odległości. W niniejszym rozdziale przedstawiono zasadę funkcjonowania poszczególnych komponentów.

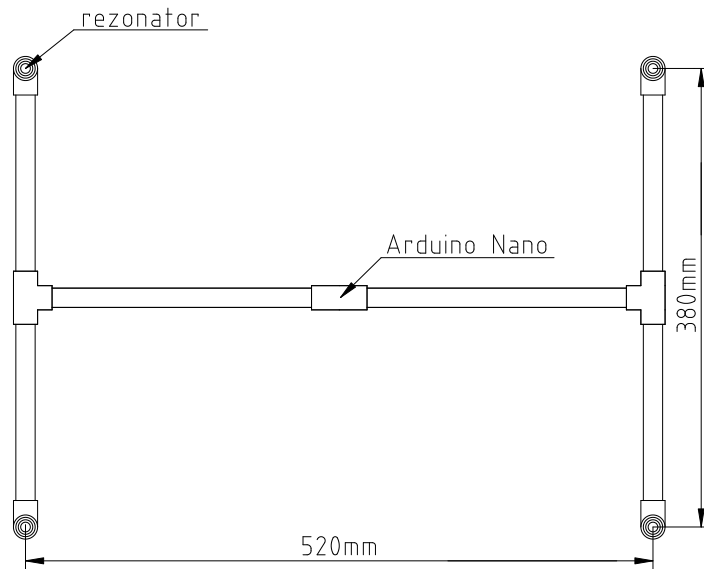
2.1. Budowa i zasada działania nadajnika

Nadajnik zbudowany został na bazie płytki prototypowej *Arduino Nano* [19], która składa się z procesora ATmega328 [13] taktowanego rezonatorem kwarcowym 16 MHz, stabilizatora napięcia 5 V oraz układu FT232RL umożliwiającego jej programowanie ze środowiska *Arduino* [20] za pośrednictwem portu USB. *Arduino Nano* połączono bezpośrednio z czterema nadajnikami ultradźwiękowymi (głośnikami, rezonatorami piezoelektrycznymi) typu 40ST-12 [30]. Za pośrednictwem złącza SV1 do płytki doprowadzono zasilanie oraz sygnały sterujące z odbiornika. Schemat połączeń przedstawia rysunek 2.1.



Rysunek 2.1: Schemat nadajnika

Całość umieszczono na ramie w kształcie litery H wykonanej z rurek PCV (rysunek 2.2). Rezonatory dodatkowo odizolowano od ramy przy użyciu rzepów, co ułatwia ich demontaż, a także skutecznie zapobiega przenoszeniu się drgań.



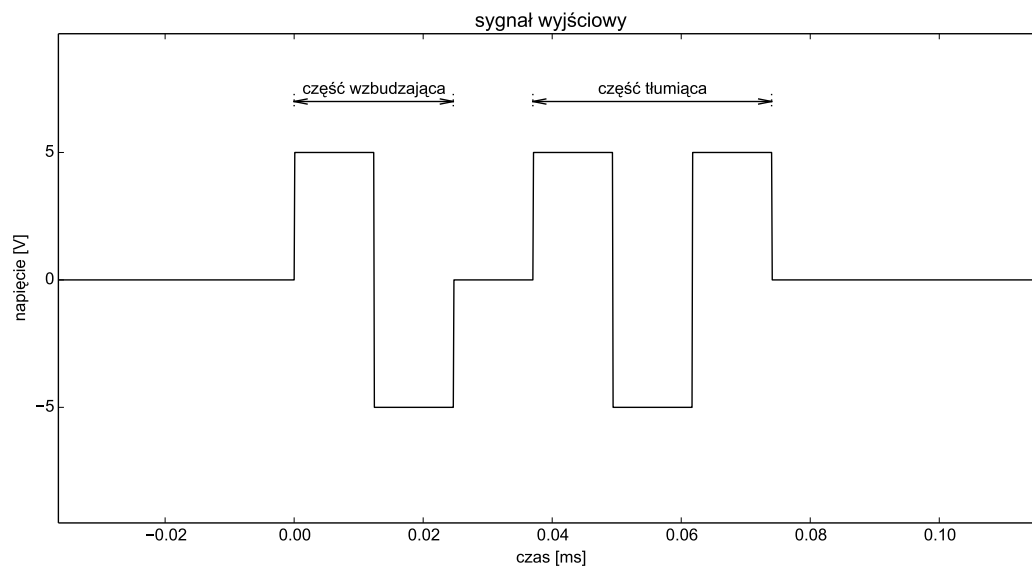
Rysunek 2.2: Szkic ramy nadajnika

Android Nano połączony jest z odbiornikiem sześciometrowym kablem, którym przesyłane są sygnały sterujące oraz zasilanie. Do sterowania wykorzystano trzy przewody – dwa z nich informują, który z głośników ma w danym momencie nadawać, a trzeci służy jako wyzwalacz. Informacja, który z głośników ma nadawać, kodowana jest za pomocą dwóch bitów w systemie binarnym – napięcie na przewodzie 3,3 V oznacza logiczną jedynkę, a 0 V oznacza logiczne zero.

Na potrzeby nadajnika powstało oprogramowanie w C dla procesora ATmega328 generujące nadawany sygnał. Cała logika programu mieści się w obsłudze przerwania sprzętowego, które reaguje na opadające zbocze na wyzwalaczu. Gdy następuje przerwanie, oprogramowanie wysyła z góry zdefiniowany sygnał do odpowiedniego rezonatora. Nadawany sygnał został tak dobrany, by dało się go w prosty sposób wyodrębnić oraz by trwał jak najkrócej. Składa się z dwóch części: wzbudzającej i tłumiącej. Okres impulsów sygnału jest zgodny z częstotliwością rezonansową przetworników, a dodatkowo część tłumiąca jest przesunięta względem części wyzwalającej o 180 stopni. Czas trwania części wzbudzającej jest najkrótszy z możliwych i trwa jeden okres, natomiast czas trwania części tłumiącej określono tak, by odebrany sygnał zawierał również sinusoidy przesunięte fazowo względem początkowej części sygnału. Rysunek 2.3 przedstawia sygnał podawany na przetwornik piezoelektryczny.

2.2. Dobór rezonatorów piezoelektrycznych

Głównym problemem podczas konstrukcji nadajnika okazał się dobór odpowiednich rezonatorów piezoelektrycznych. Mimo że producent wykorzystanych rezonatorów zapewnia ich pracę w zakresie $40\text{ kHz} \pm 1\text{ kHz}$, to taki rozrzut okazał się zbyt duży, dlatego z 30 rezonatorów (15



Rysunek 2.3: Sygnał podawany na przetwornik piezoelektryczny

nadajników i 15 odbiorników) wybrane zostały 4 nadajniki i 3 odbiorniki o najbardziej zbliżonych częstotliwościach rezonansowych. Tabela 2.1 zawiera wyniki pomiarów częstotliwości. Gwiazdką oznaczono wykorzystane przetworniki piezoelektryczne.

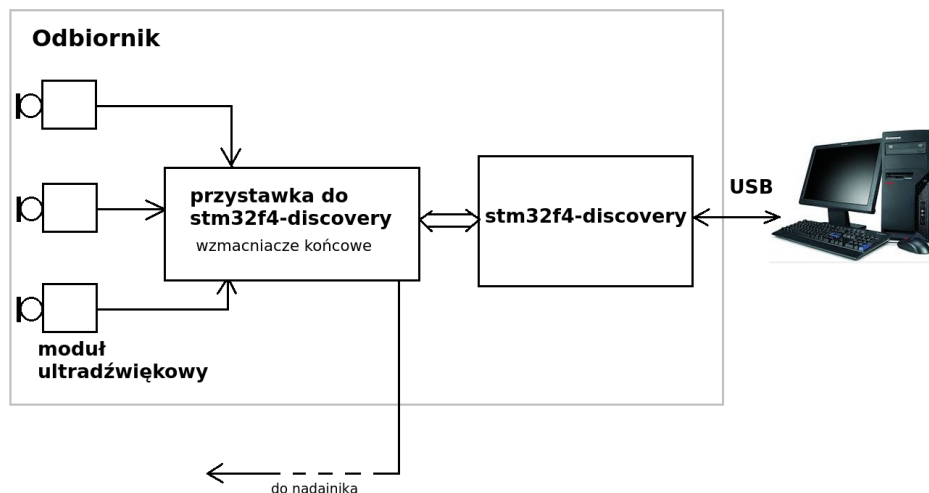
Tabela 2.1: Częstotliwości rezonansowe przetworników piezoelektrycznych

Nr	Nadajnik 40ST-12	Odbiornik 40SR-12
1	40,88 kHz	*40,65 kHz
2	41,12 kHz	40,45 kHz
3	*40,78 kHz	39,52 kHz
4	41,19 kHz	40,47 kHz
5	40,92 kHz	40,66 kHz
6	39,68 kHz	*40,69 kHz
7	39,78 kHz	40,59 kHz
8	*40,80 kHz	40,39 kHz
9	40,90 kHz	40,29 kHz
10	*40,66 kHz	*40,68 kHz
11	*40,85 kHz	39,22 kHz
12	41,01 kHz	39,51 kHz
13	41,00 kHz	39,92 kHz
14	39,82 kHz	39,26 kHz
15	39,64 kHz	39,11 kHz

2.3. Odbiornik

Centralną część prezentowanego prototypu stanowi odbiornik. Jego zadaniem jest wysyłanie sygnałów sterujących do nadajnika, zbieranie ultradźwięków z otoczenia oraz przesyłanie ich do komputera w celu dalszej analizy. Odbiornik składa się z następujących części (rysunek 2.4):

- trzech modułów ultradźwiękowych przetwarzających dźwięk na sygnał elektryczny,
- płytki prototypowej *stm32f4-discovery* [18] odpowiadającej za komunikację z komputerem,
- przystawki do *stm32f4-discovery* przystosowującej sygnały elektryczne z modułów ultradźwiękowych do poziomów akceptowalnych przez tę płytkę, z przystawki wyprowadzono również sygnały sterujące nadajnikiem,
- ramy, na której umieszczono moduły ultradźwiękowe.



Rysunek 2.4: Szkic odbiornika

2.4. Budowa i zasada działania odbiornika

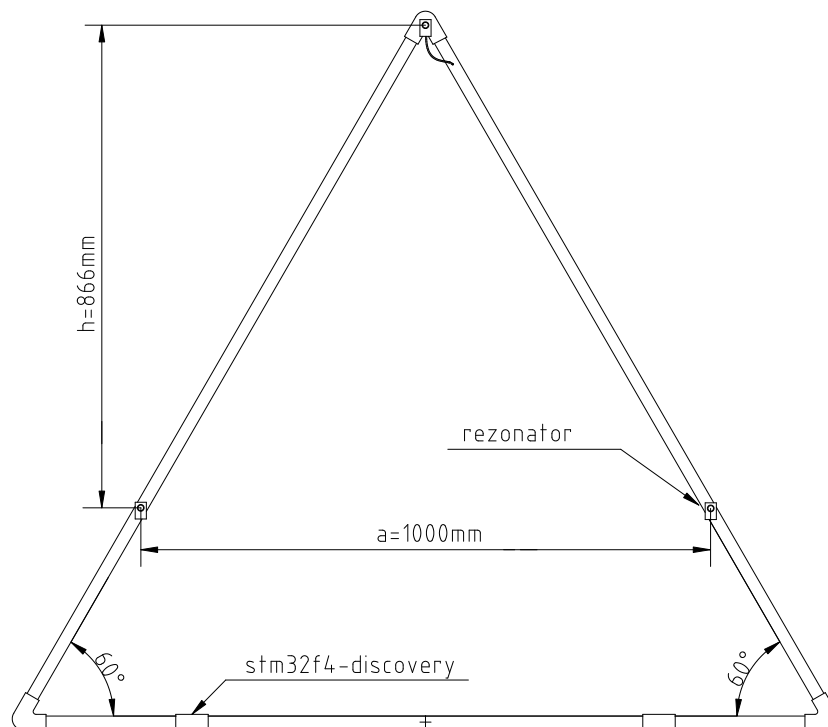
Głównym elementem odbiornika jest płytki prototypowa *stm32f4-discovery* [18]. Umożliwia ona komunikację wszystkich komponentów z komputerem. *Stm32f4-discovery* wyposażona jest w procesor STM32F407VGT6 [14] (oparty o rdzeń ARM Cortex M4), który zawiera trzy 12-bitowe przetworniki analogowo-cyfrowe umożliwiające próbkowanie z prędkością do 2,4 MSPS (ang. *MSPS* – *Mega-Samples Per Second* – milion próbek na sekundę). Przetworniki te wykorzystane zostały do próbkowania sygnałów pochodzących z modułów ultradźwiękowych. Płytki umożliwia również komunikację z komputerem przez port USB z prędkością 12 MB/s. Znajduje się na niej także programator umożliwiający programowanie procesora za pośrednictwem dodatkowego portu USB.

Na potrzeby odbiornika powstało dedykowane oprogramowanie w C sterujące procesorem. Zostało ono oparte na bibliotece *stm32 usb 101* [9] zapewniającej komunikację z komputerem.

Do biblioteki dodano obsługę przetworników analogowo-cyfrowych oraz możliwość sterowania nadajnikiem. Program przez port USB dostaje instrukcję, który z czterech głośników ma nadawać, i przekazuje ją dalej do nadajnika wraz z sygnałem wyzwalającym. Następnie uruchamiane są równocześnie trzy przetworniki analogowo-cyfrowe, które próbują odbierać dźwięk i poprzez DMA zapisują wynik w pamięci procesora. Częstotliwość pracy przetworników ustawiono na 1,6 MSPS, co daje 40 próbek na jeden okres sygnału o częstotliwości 40 kHz. Po zebraniu w sumie $3 \times 16k = 48k$ próbek całość przesyłana jest do komputera w celu dalszej analizy. Proces powtarza się dla każdego z czterech głośników nadajnika, co w sumie daje 12 sygnałów, na podstawie których wyznaczona zostaje pozycja w przestrzeni oraz orientacja nadajnika.

Całą elektronikę osadzono na ramie w kształcie trójkąta zbudowanej z rur PCV (rysunek 2.5). Odległości pomiędzy modułami ultradźwiękowymi są z góry ustalone, co ułatwia dalsze obliczenia.

Warto zaznaczyć, że dla dźwięku poruszającego się z prędkością 340 m/s długość wykorzystanej fali dźwiękowej wynosi 8,5 mm, co przy pomiarze 40 próbek na jeden okres daje rozdzielczość pomiaru odległości mniejszą niż 0,5 mm (dokładnie 0,2125 mm). Błąd pomiaru trudno jednak oszacować, ponieważ zależy on od aktualnej prędkości dźwięku, a ta zmienia się w czasie wraz ze zmianą warunków atmosferycznych.

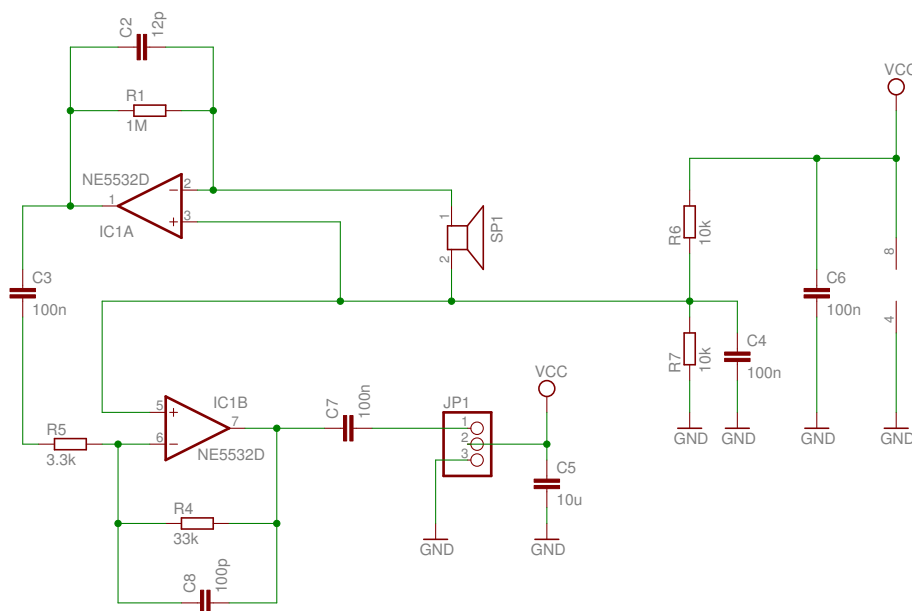


Rysunek 2.5: Szkic ramy odbiornika

2.5. Budowa modułu ultradźwiękowego

Odbiornik wyposażono w trzy moduły ultradźwiękowe, których zadaniem jest zbieranie ultradźwięków z trzech różnych punktów. Każdy moduł zawiera przetwornik piezoelektryczny

(mikrofon) 40SR-12 [30], który przetwarza sygnał akustyczny na odpowiadający mu sygnał elektryczny, oraz wzmacniacze operacyjne wstępnie zwiększające amplitudę sygnału, który przesyłany jest dalej do przystawki. Rysunek 2.6 przedstawia schemat modułu ultradźwiękowego.



Rysunek 2.6: Schemat modułu ultradźwiękowego

Wzmacniacz operacyjny IC1A wraz z kondensatorem C2 i rezystorem R1 pracuje jako przedwzmacniacz ładunkowy [6]. Ładunek wytworzony na przetworniku piezoelektrycznym SP1 zostaje w całości przeniesiony na kondensator C2 (wzmacniacz utrzymuje różnicę potencjałów między dodatnim a ujemnym wejściem na zerowym poziomie). W rezultacie na kondensatorze C2 pojawia się napięcie proporcjonalne do wartości odebranego sygnału. Do rozładowania kondensatora C2 służy rezystor R1. R1 i C2 działają również jako filtr dolno-przepustowy.

Wzmacniacz IC1B z rezystorami R5 i R4 oraz kondensatorami C3, C7 i C8 pracuje jako wzmacniacz napięciowy wzbogacony o filtr pasmowoprzepustowy. Sygnał z IC1B za pośrednictwem wtyczki JP1 dociera do przystawki współpracującej z *stm32f4-discovery*.

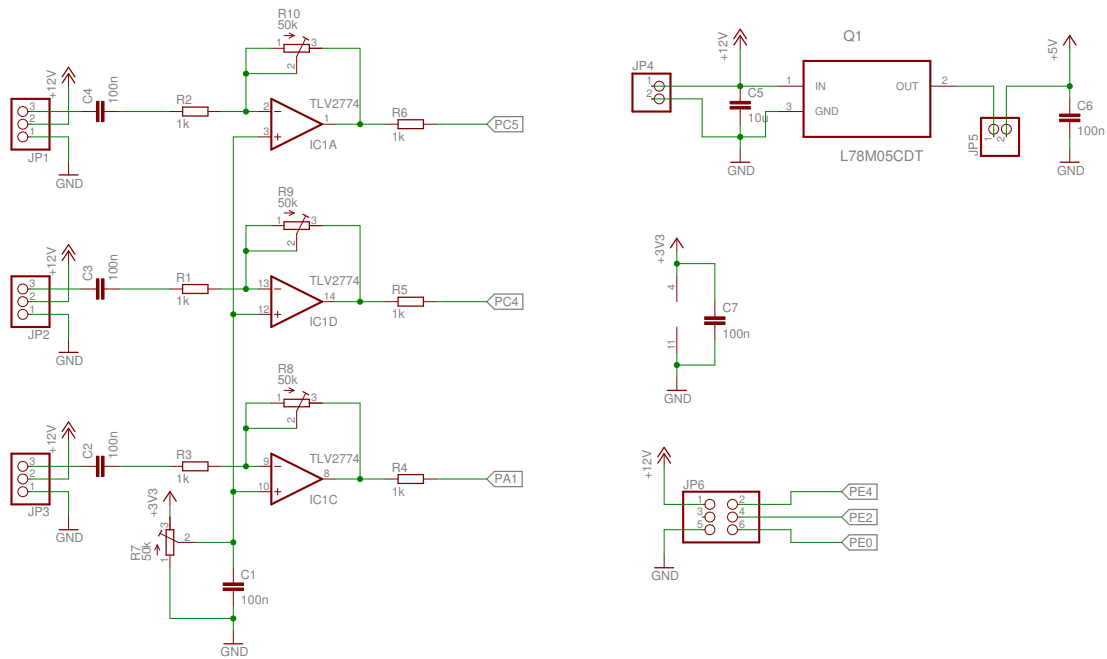
W celu zminimalizowania zakłóceń zastosowano niskoszumowe wzmacniacze operacyjne mieszczące się w jednym układzie scalonym NE5532 [15]. Dodatkowo płytką drukowaną jest ekranowana.

2.6. Przystawka do *stm32f4-discovery*

Sygnał z modułów ultradźwiękowych dociera do *stm32f4-discovery* za pośrednictwem specjalnej przystawki. Schemat budowy przystawki przedstawia rysunek 2.7. Przystawka przystosowuje maksymalne amplitudy zebranych sygnałów do wartości akceptowalnych przez przetworniki analogowo-cyfrowe procesora STM32F407VGT6. Wartości te muszą mieścić się w zakresie od 0 V do 3,3 V.

Zastosowano w niej układ TLV2774 [16], który zawiera 4 wzmacniacze operacyjne typu *rail-to-rail*. Trzy z nich wykorzystano jako ostatni stopień wzmocnienia sygnałów ultradźwiękowych. Wzmacniacze operacyjne pracują w układzie odwracającym, którego wzmocnienie

można regulować odpowiednio potencjometrami R8, R9 oraz R10. Podłączono je do wspólnego, również regulowanego (przy użyciu potencjometru R7) napięcia odniesienia. Przystawka zawiera także stabilizator napięcia LM78M05CDT [17], który po podłączeniu do JP4 baterii 12V dostarcza zasilanie do wszystkich komponentów. Istnieje możliwość odcięcia zasilania przez rozwarcie JP5, co jest konieczne podczas programowania STM32F407VGT6, by zapobiec uszkodzeniu stabilizatora napięcia. Z przystawki przez wtyczkę JP6 wyprowadzono również sygnały sterujące nadajnikiem oraz zasilanie. Wtyczka JP6 połączona jest z wtyczką SV1 nadajnika (rysunek 2.1) sześciometrowym kablem.



Rysunek 2.7: Schemat budowy przystawki do *stm32f4-discovery*

Rozdział 3

Przetwarzanie danych zebranych z odbiornika

W ramach niniejszej pracy powstało oprogramowanie, którego zadaniem jest przekształcenie surowych danych ultradźwiękowych z odbiornika na informację o położeniu nadajnika w przestrzeni oraz jego orientacji. Na oprogramowanie składają się biblioteka *mp3d* zajmująca się sterowaniem odbiornikiem, odbieraniem sygnałów ultradźwiękowych i ich analizą, program *scan.py* wizualizujący dane w postaci obrazu trójwymiarowego oraz program *save-pattern.py*, który służy do wstępnej kalibracji urządzenia.

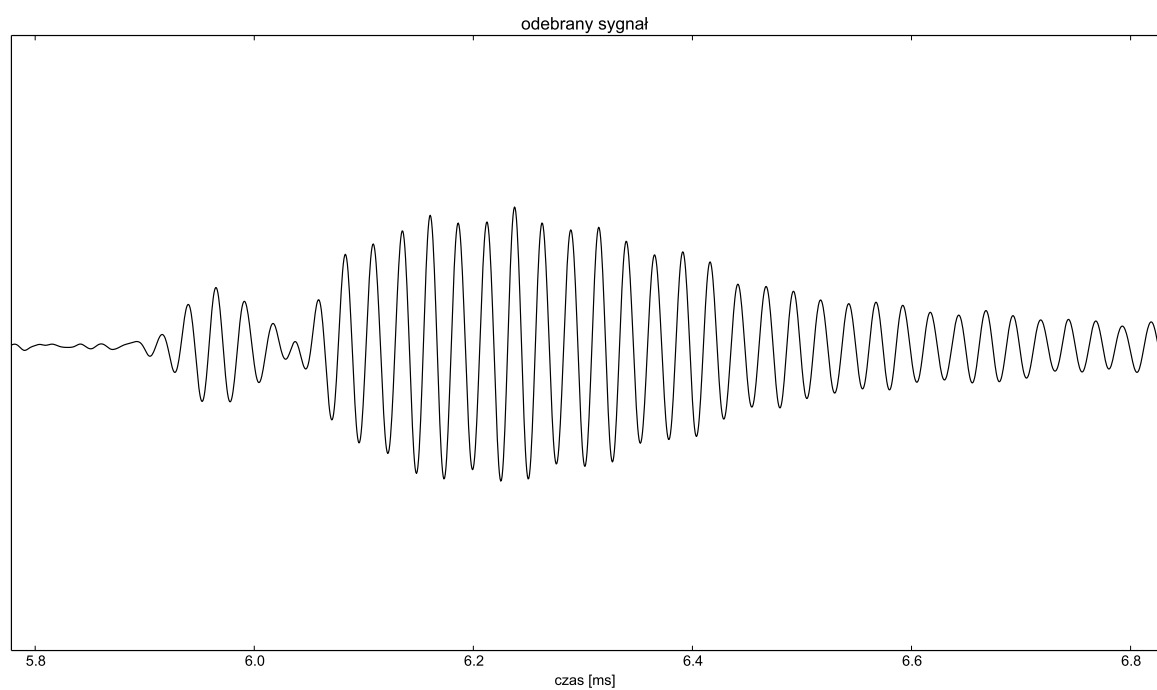
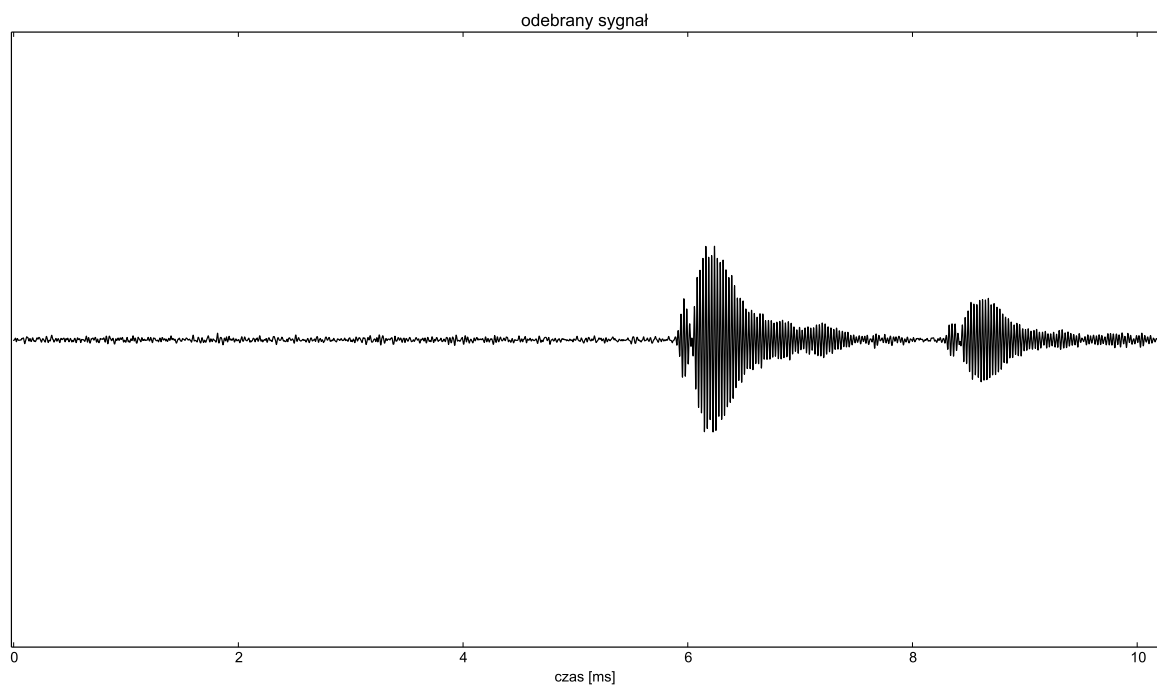
Najistotniejszą częścią oprogramowania jest biblioteka *mp3d* podzielona na pięć modułów:

- *com.py* – odpowiedzialny za komunikację z odbiornikiem,
- *find-pattern.py* – odpowiedzialny za wyznaczanie odległości przez wyszukanie wzorca w odebranym sygnale,
- *xyz.py* – odpowiedzialny za wyznaczenie pozycji i orientacji nadajnika oraz weryfikujący, czy zebrane dane odpowiadają modelowanej rzeczywistości,
- *info.py* – odpowiedzialny za wyświetlanie informacji o sile odbieranego sygnału,
- *ply.py* – odpowiedzialny za eksportowanie danych do formatu *.ply* (*ang. Polygon File Format*) obsługiwanego przez większość programów do obróbki grafiki 3D.

3.1. Komunikacja z odbiornikiem, moduł *com.py*

Za komunikację z odbiornikiem odpowiada moduł *com.py*. Pracuje on w oddzielnym wątku, w którym cyklicznie wysyłane są żądania, by dany głośnik nadał sygnał, a następnie odbierane są sygnały z trzech mikrofonów. Ten proces odbywa się sukcesywnie dla każdego z czterech głośników nadajnika. Odstęp pomiędzy kolejno nadawanymi sygnałami wynosi 87 ms, co pozwala na całkowite rozproszenie się fali dźwiękowej, aby kolejne fale nie nakładały się na siebie.

Zebrane w ten sposób dwanaście sygnałów po wstępnej filtracji przekazywane jest dalej do *find-pattern.py*. Cały cykl powtarza się co 350 ms. Rysunek 3.1 przedstawia sygnał odebrany z jednego z mikrofonów. Odebrany sygnał znacząco różni się od nadanego (rysunek 2.3), ponieważ wykorzystane mikrofony i głośniki są rezonatorami, a dodatkowo ich obudowy tworzą pudła rezonansowe.



Rysunek 3.1: Sygnał odebrany przez moduł *com.py*. Odległość między nadajnikiem a odbiornikiem wynosi 2 m. Na górnym wykresie w przedziale 6 ms i 8 ms widać właściwy sygnał (poruszający się po linii prostej), a od 8,5 ms widać sygnał odbity od podłogi. Dolny wykres jest powiększeniem fragmentu górnego wykresu.

3.2. Wyznaczanie odległości, moduł *find_pattern.py*

Moduł *find_pattern.py* odpowiada za obliczanie odległości pomiędzy czterema głośnikami i trzema mikrofonami. Wyznaczana jest ona przez wyszukanie w każdym z 12 odebranych sygnałów położenia *wzorca*, który odpowiada czołu nadanego sygnału (rysunek 3.1). Wzorzec po raz pierwszy wprowadzana się podczas kalibracji, a podczas pracy urządzenia jest on ciągle aktualizowany.

Aby wyszukać wzorzec wewnątrz sygnału, korzysta się z metody najmniejszego błędu średniokwadratowego. Niech $w(t)$ dla $t = 0, \dots, n-1$ będzie szukany wzorcem, a $f(t)$ dla $t = 0, \dots, m-1$ odebrany sygnałem. Wtedy możemy znaleźć takie a , że błąd średniokwadratowy $E(t)$ pomiędzy w i $af|_{\{t, \dots, t+n-1\}}$ jest minimalny. Możemy to zapisać w postaci

$$E(t) = \min_{a \in R} \left\{ \sum_{i=0}^{n-1} (w(i) - af(t+i))^2 \right\} \quad \text{dla } t = 0, \dots, m-n.$$

Po podniesieniu wyrażenia w nawiasie do kwadratu otrzymujemy

$$E(t) = \min_{a \in R} \left\{ \sum_{i=0}^{n-1} (w^2(i) - 2aw(i)f(t+i) + a^2f^2(t+i)) \right\},$$

$$E(t) = \min_{a \in R} \left\{ \sum_{i=0}^{n-1} w^2(i) - 2a \sum_{i=0}^{n-1} w(i)f(t+i) + a^2 \sum_{i=0}^{n-1} f^2(t+i) \right\}.$$

Aby wyliczyć $E(t)$, należy zminimalizować wyrażenie

$$y(a) = \sum_{i=0}^{n-1} w^2(i) - 2a \sum_{i=0}^{n-1} w(i)f(t+i) + a^2 \sum_{i=0}^{n-1} f^2(t+i).$$

Korzystając z faktu, że $y(a)$ jest funkcją kwadratową, możemy wyliczyć a minimalizujące $y(a)$

$$a = \frac{\sum_{i=0}^{n-1} w(i)f(t+i)}{\sum_{i=0}^{n-1} f^2(t+i)}.$$

Ostatecznie dostajemy wzór na $E(t)$

$$E(t) = \sum_{i=0}^{n-1} w^2(i) - \frac{\left(\sum_{i=0}^{n-1} w(i)f(t+i) \right)^2}{\sum_{i=0}^{n-1} f^2(t+i)}.$$

Zauważmy, że dzięki skalowaniu skalarą a sygnału f , zamiast wzorca w , otrzymane $E(t)$ nie zależy od siły odebranego sygnału. Ułatwia to porównanie błędów w dwóch różnych miejscach. $E(t)$ zależy jednak od siły sygnału wzorcowego – możemy się od niej uniezależnić, wyznaczając błąd względny $e(t)$

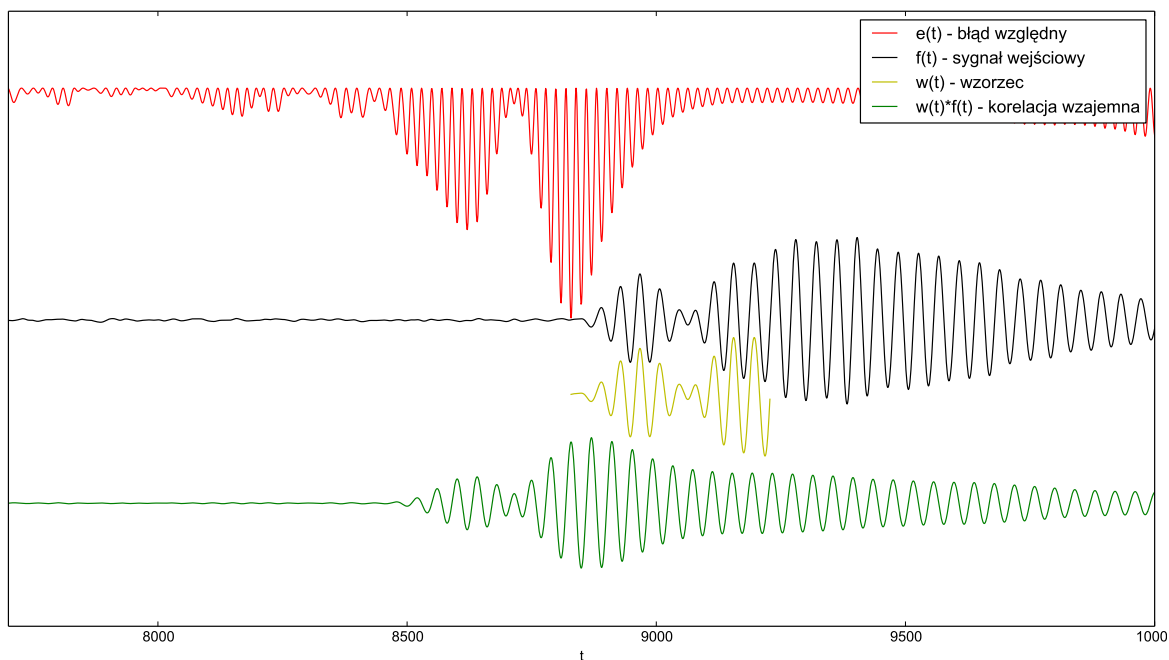
$$e(t) = \frac{E(t)}{\sum_{i=0}^{n-1} w^2(i)}.$$

Po podstawieniu $E(t)$ dostajemy

$$e(t) = 1 - \frac{\left(\sum_{i=0}^{n-1} w(i)f(t+i) \right)^2}{\sum_{i=0}^{n-1} f^2(t+i) \sum_{i=0}^{n-1} w^2(i)}.$$

Funkcję $e(t)$ można interpretować następująco: im mniejszy błąd $e(t)$, tym większe prawdopodobieństwo, że szukany wzorec w znajduje się na pozycji t w odebranym sygnale f .

Wynikiem obliczeń modułu *find_pattern.py* jest cała funkcja $e(t)$, na podstawie której moduł *xyz.py* wyznacza pozycję i orientację nadajnika, uwzględniając przy tym jego kształt (nadmiarowość danych), a także prawdopodobieństwo, że znaleziony wzorec znajduje się w określonej pozycji. Rysunek 3.2 przedstawia wynik przetwarzania sygnału przez moduł *find_pattern.py*.



Rysunek 3.2: Sygnał przetworzony przez moduł *find_pattern.py*

Wyliczenie zarówno $\sum_{i=0}^{n-1} w^2(i)$, jak i $\sum_{i=0}^{n-1} f^2(t+i)$ wymaga jedynie liniowej liczby operacji, a $\sum_{i=0}^{n-1} w(i)f(t+i)$ jest korelacją wzajemną funkcji $w(t)$ i $f(t)$, którą można wyliczyć w czasie $k \log(k)$, korzystając z szybkiej transformacji Fouriera [11], gdzie $k = \max\{n, m\}$. Wyliczenie całej funkcji $e(t)$ wymaga zatem jedynie $k \log(k)$ operacji.

Kolejne zadanie modułu *find_pattern.py* to uaktualnianie wzorca. Wraz ze zmianą kąta nachylenia nadajnika względem odbiornika zmienia się kształt odbieranego sygnału, dlatego jeśli badany sygnał $f(t)$ zawiera szukany wzorec na pozycji t_w oraz spełnione są następujące warunki:

- moc sygnału $f(t_w), \dots, f(t_w + n - 1)$ jest wystarczająco duża, gdzie n jest długością wzorca,
- błąd względny $e(t_w)$ pomiędzy sygnałem a wzorcem jest mały,
- moduł *xyz.py* rozpozna otrzymaną pozycję t_w jako pasującą do modelowanej rzeczywistości

to moduł uaktualni wzorec do wartości $f(t_w), \dots, f(t_w + n - 1)$.

3.3. Wyznaczanie pozycji oraz orientacji przestrzennej, moduł *xyz.py*

Na ostatnim etapie obliczeń wyznaczane są współrzędne kartezjańskie czterech głośników umieszczonych na nadajniku. Zajmuje się tym moduł *xyz.py*, który dodatkowo zawiera prosty model rzeczywistości rozpoznający niepoprawne dane o położeniu głośników. Model zawiera informacje o wzajemnym położeniu czterech głośników w przestrzeni, co umożliwia wychwytywanie zmiany geometrii nadajnika (niepoprawne odległości pomiędzy głośnikami).

Współrzędne głośników wyznacza się na podstawie dwunastu funkcji $e_{ij}(t)$ dla $i = 1, \dots, 4$ oraz $j = 1, \dots, 3$, w których zawarta jest informacja o odległości pomiędzy głośnikiem i oraz mikrofonem j . Dla danej funkcji $e_{ij}(t)$ wyszukiwane są wszystkie wartości t_m , dla których funkcja osiąga minimum lokalne. Następnie z t_m wybierane są trzy punkty $t_{m_1}, t_{m_2}, t_{m_3}$, dla których wartość $e_{ij}(t_m)$ jest najmniejsza, czyli spełniony został warunek:

$$e_{ij}(t_{m_1}) \leq e_{ij}(t_{m_2}) \leq e_{ij}(t_{m_3}) \leq e_{ij}(t_m) \quad \text{dla } m \neq m_1, m_2, m_3.$$

Pozycje $t_{m_1}, t_{m_2}, t_{m_3}$ można interpretować jako trzy najbardziej prawdopodobne odległości pomiędzy głośnikiem a mikrofonem.

Do wyznaczenia położenia głośników w przestrzeni wykorzystywane są wszystkie możliwe kombinacje odległości $t_{m_1}, t_{m_2}, t_{m_3}$ dla każdej z dwunastu funkcji e_{ij} . Daje to $12^3 = 1728$ możliwych lokalizacji głośników w przestrzeni. Każdą z nich sprawdza się z modelem (uzyskana geometria przestrzenna nadajnika musi zgadzać się z modelem). Z tych, które pasują, wybierana jest lokalizacja o najmniejszym sumarycznym błędzie średniokwadratowym.

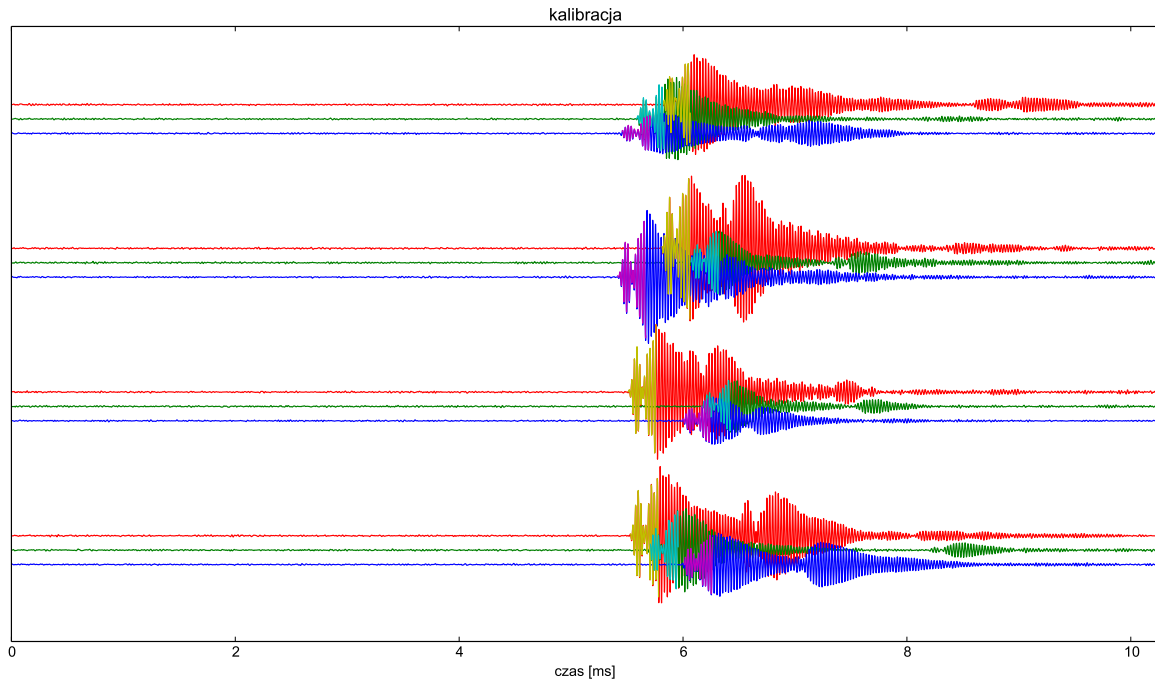
Ostatecznym wynikiem pracy modułu *xyz.py* jest dwanaście wartości t_{m_k} gdzie $k \in \{1, 2, 3\}$, dla każdej funkcji e_{ij} . Wartości te określają pozycję głośników w przestrzeni.

Dodatkowo wartości t_{m_k} przekazywane są z powrotem do modułu *find_pattern.py* w celu aktualizacji wzorców.

3.4. Kalibracja

Po każdym uruchomieniu urządzenie wymagana kalibracji, która polega na ustawieniu nadajnika w odległości 2 metrów od odbiornika, uruchomieniu programu *save-pattern.py* oraz zaznaczeniu w nim 12 obszarów będącym czołem odebranego sygnału. Obszary te wykorzystane zostaną jako wzorce podczas mierzenia odległości. Program wyliczy także aktualną prędkość dźwięku, uwzględniając odległość nadajnika od odbiornika.

Rysunek 3.3 przedstawia 12 sygnałów wraz z zaznaczonymi wzorcami.



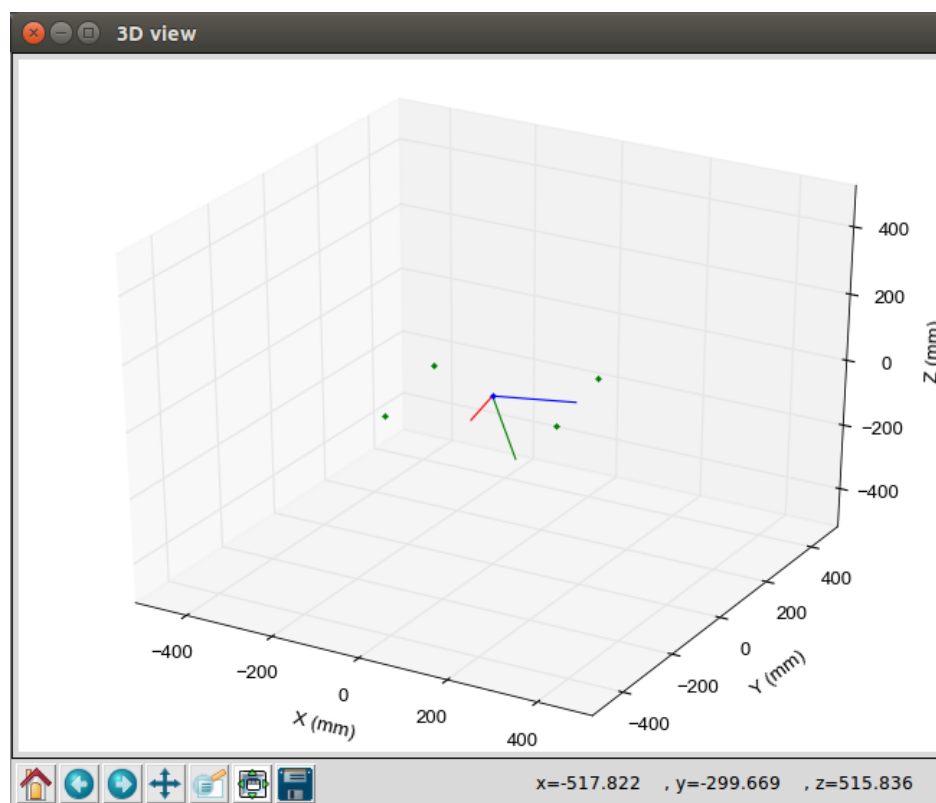
Rysunek 3.3: Kalibracja, 12 sygnałów z zaznaczonymi wzorcami

3.5. Obsługa programu *scan.py*

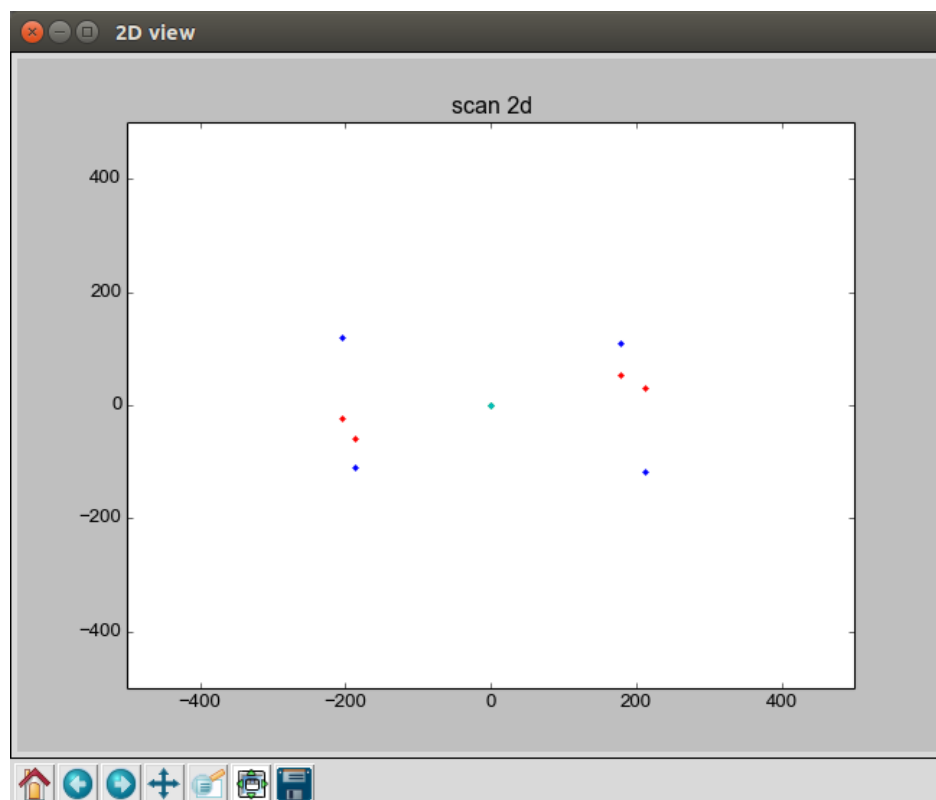
Do obsługi prototypu służy program *scan.py*. Po jego uruchomieniu na monitorze wyświetlają się trzy okna:

- widok 3D prezentujący położenie nadajnika w przestrzeni – w oknie widoczne są 4 punkty reprezentujące głośniki nadajnika oraz 3 prostopadłe względem siebie wektory określające jego orientację i położenie (rysunek 3.4),
- widok 2D, również przedstawiający położenie nadajnika w przestrzeni – są to nałożone na siebie rzuty na płaszczyzny XY i XZ (rysunek 3.5),
- diagram informujący o mocy całego odebranego sygnału, mocy wzorca, mocy sygnału pasującego do wzorca oraz błędzie pomiędzy wzorcem a odebranym sygnałem (rysunek 3.6).

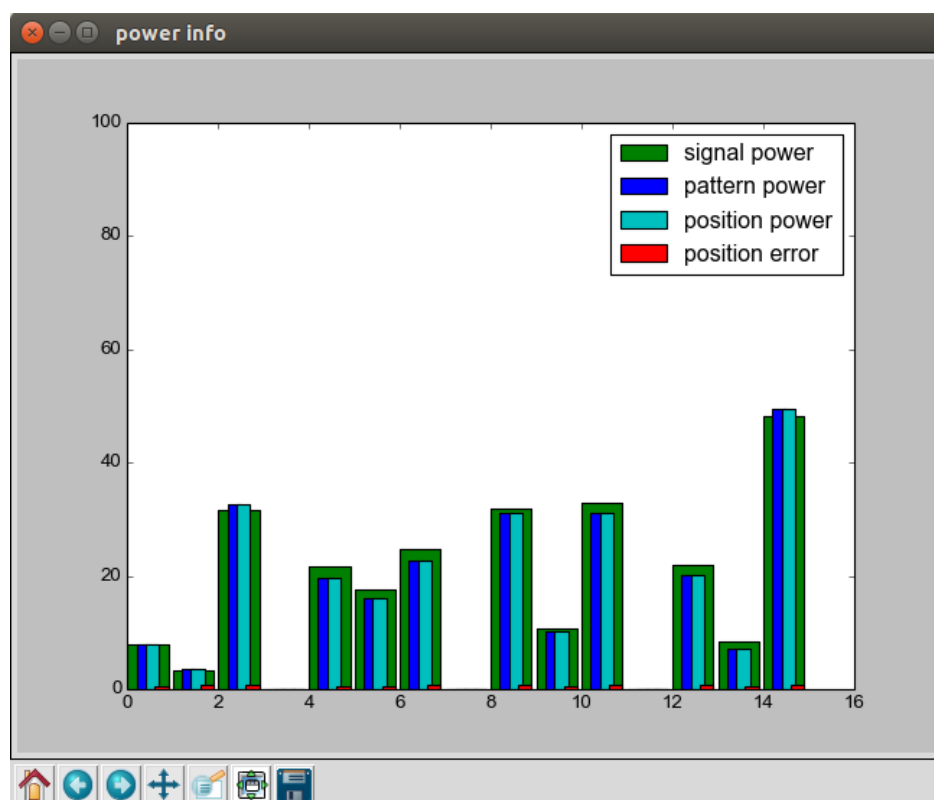
Wszelkie zmiany położenia nadajnika automatycznie odświeżają wszystkie widoki. Podczas uruchamiania programu aktualne położenie nadajnika przyjmowane jest jako początek układu współrzędnych. Jeśli któryś z głośników straci widoczność z mikrofonami, obraz przestaje się odświeżać, a informacja, dlaczego tak się dzieje, widoczna jest w oknie przedstawiającym moc sygnału.



Rysunek 3.4: Widok 3D



Rysunek 3.5: Widok 2D



Rysunek 3.6: Diagram informujący o mocy całego odebranego sygnału, mocy wzorca, mocy sygnału pasującego do wzorca oraz błędzie pomiędzy wzorcem a odebrany sygnałem

Rozdział 4

Podsumowanie

W niniejszej pracy przedstawiono urządzenie, które za pomocą ultradźwięków potrafi określić zarówno swoje położenie, jak i orientację w przestrzeni.

Mimo że przy pomiarze położenia wykorzystywane są fale dźwiękowe o długość 8,5 mm, urządzenie to potrafi osiągnąć rozdzielczość rzędu 0,5 mm, co znacząco wyróżnia je na tle innych rozwiązań. Posiada także inne zalety, jak również wady, które opisano poniżej.

Zalety:

- Wysoka rozdzielczość pomiaru – zastosowanie częstotliwości próbkowania dużo wyższej od częstotliwości nadawanego sygnału, a także wyszukiwanie wzorca pozwoliły wielokrotnie zwiększyć rozdzielczość pomiaru w stosunku do długości wykorzystanej fali dźwiękowej.
- Praktycznie nieograniczona przestrzeń robocza – mimo że w prezentowanym prototypie ograniczono przestrzeń roboczą do sześcianu $2,5\text{ m} \times 2,5\text{ m} \times 2,5\text{ m}$, jest to jedynie ograniczenie praktyczne i bez trudu można dowolnie zwiększyć tę przestrzeń. Trzeba jednak pamiętać, że siła sygnału dźwiękowego maleje kwadratowo wraz z odległością, więc przy dużo większych odległościach niezbędne staje się zastosowanie nadajników większej mocy.
- Niski koszt komponentów.

Wady:

- Duże gabaryty nadajnika i odbiornika.
- Duża czułość na zmiany temperatury – jest to najpoważniejsza wada prezentowanego rozwiązania. Zauważmy, że zmiana temperatury otoczenia o 1°C powoduje zmianę prędkości dźwięku o 0,17%, co przy rozdzielczości 5000 px w skrajnych przypadkach daje przesunięcie 7,5 px. Co prawda w warunkach laboratoryjnych nie stanowi to większego problemu, jednak w praktycznych zastosowaniach jest poważnym ograniczeniem.
- Długi czas odświeżania – prezentowany prototyp dokonuje pomiarów co 350 ms, czyli około 3 pomiarów na sekundę. Ponieważ mierzymy cztery niezależne sygnały (z czterech głośników), poszczególne sygnały mierzone są co 87 ms. Dźwięk poruszający się z prędkością 346 m/s przebędzie w tym czasie odległość 29,7 m. Tak duży dystans w stosunku do powierzchni roboczej jest konieczny, by zapobiec nakładaniu się sygnałów (które po odbiciu się od ścian mogłyby wpłynąć na kolejne pomiary; czas 87 ms pozwala na ich całkowite rozproszenie). Zakłócaniu pomiarów można także zaradzić, nadając za każdym

razem inny rozróżnialny sygnał, to podejście wymaga jednak nadajników i odbiorników o większych możliwościach sterowania.

- Stosunkowo mały kąt widzenia – zastosowane przetworniki piezoelektryczne są przetwornikami kierunkowymi, co ogranicza maksymalny kąt, pod jakim można ustawić nadajnik względem odbiornika. W prezentowanym prototypie kąt ten nie powinien przekraczać 45° . Zastosowanie większej liczby przetworników lub innego ich rodzaju pozwoliłoby znacznie zwiększyć kąt.
- Ograniczone możliwościysterowania rezonatorów piezoelektrycznych – zastosowane rezonatory nie nadają się do generowania krótkich impulsów, co zdecydowanie uprościłoby mierzenie odległości.
- Problem z ustaleniem pozycji i orientacji nadajnika w ruchu – w prezentowanym urządzeniu jeden pomiar pozycji i orientacji wymaga określania położenia czterech głośników w różnych momentach czasu. Jeśli podczas takiego pomiaru nadajnik porusza się ze zbyt dużą prędkością, to otrzymamy położenia przesunięte względem siebie zgodnie z kierunkiem ruchu nadajnika. Takie przesunięcie negatywnie wpływa na pomiar orientacji.

Reasumując, prezentowane urządzenie, wykorzystujące ultradźwięki do określania pozycji i orientacji, jest innowacyjną propozycją, która zapewnia znacznie większą dokładność niż wszystkie inne dostępne rozwiązania. Najistotniejszą jego wadą wydaje się jednak duża wrażliwość na warunki atmosferyczne, a szczególnie na zmianę temperatury. Jest to zapewne główny powód, dla którego rozwiązanie to nie znajdzie szerokiego zastosowania w produktach komercyjnych.

Dodatek A

Załączniki

Na płycie CD-ROM dołączonej do niniejszej pracy znajdują się:

- kopia niniejszej pracy w PDF i jej tekst źródłowy w języku \LaTeX ,
- kod źródłowy nadajnika,
- kod źródłowy odbiornika,
- kod źródłowy programu *scan.py*
- kod źródłowy programu do kalibracji *save-pattern.py*
- kod źródłowy biblioteki *mp3d*, na którą składają się:
 - moduł *mp3d/com.py*
 - moduł *mp3d/find_pattern.py*
 - moduł *mp3d/info.py*
 - moduł *mp3d/ply.py*
 - moduł *mp3d/xyz.py*

Bibliografia

- [1] Banker C., Cretella M., Dimaria J., Mitchell J., Tucker J., *Ultrasonic 3D Wireless Computer Mouse*, Worcester Polytechnic Institute Electrical and Computer Engineering Department, kwiecień 2007
https://www.wpi.edu/Pubs/E-project/Available/E-project-042607-125024/unrestricted/magicmouse_2007-04-26.pdf
- [2] Changchang W., *VisualSFM: A Visual Structure from Motion System*
<http://ccwu.me/vsfm/>
- [3] Dean E.A., *Atmospheric Effects on the Speed of Sound*, BATTELLE COLUMBUS LABS DURHAM NC, sierpień 1979
<http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=AD A076060>
- [4] Desai P.R., Desai P.N., AV Ajmera K.D., Mehta K., *A Review Paper on Oculus Rift – A Virtual Reality Headset*
<http://arxiv.org/pdf/1408.1173.pdf>
- [5] Hua J., Shen Z., Zhong S., *We Can Track You If You Take the Metro: Tracking Metro Riders Using Accelerometers on Smartphones*
<http://arxiv.org/pdf/1505.05958v1.pdf>
- [6] Karki J., *Signal Conditioning Piezoelectric Sensors*, Application Report, SLOA033A, wrzesień 2000
<http://www.ti.com/lit/an/sloa033a/sloa033a.pdf>
- [7] Markham A., Trigoni N., Ellwood S.A., Macdonald D.W., *Revealing the hidden lives of underground animals using magneto-inductive tracking*
<https://www.cs.ox.ac.uk/files/3202/sensys2010b.pdf>
- [8] Narayana S., Prasad R.V., Rao V.S., Prabhakar T.V., Kowshik S.S., Iyer M.S., *PIR Sensors: Characterization and Novel Localization Technique*
<http://www.es.ewi.tudelft.nl/papers/2015-VP-PIR.pdf>
- [9] Peczarski M., *USB dla niewtajemniczonych w przykładach na mikrokontrolery STM32*, Wydawnictwo BTC, Legionowo 2013
http://www.mimuw.edu.pl/~marpe/book/stm32_usb_101.zip
- [10] Sapiezynski P., Stopczynski A., Gatej R., Lehmann S., *Tracking Human Mobility using WiFi signals*
<http://arxiv.org/pdf/1505.06311v1>

- [11] Weisstein E.W., *Cross-Correlation Theorem*. From MathWorld – A Wolfram Web Resource
<http://mathworld.wolfram.com/Cross-CorrelationTheorem.html>
- [12] Zeng W., *Microsoft Kinect Sensor and Its Effect*, Multimedia at Work, grudzień 2012
<http://research.microsoft.com/en-us/um/people/zhang/Papers/Microsoft%20Kinect%20Sensor%20and%20Its%20Effect%20-%20IEEE%20MM%202012.pdf>
- [13] Atmel-8271I, *ATmega48A/PA/88A/PA/168A/PA/328/P – ATMEL 8-BIT MICROCONTROLLER WITH 4/8/16/32KBYTES IN-SYSTEM PROGRAMMABLE FLASH*, październik 2014
http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf
- [14] RM0090 Reference manual, *STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced ARM®-based 32-bit MCUs*
<http://www.st.com/stm32>
- [15] SLOS075I, *Dual low-noise operational amplifiers*, listopad 1979 – kwiecień 2009
<http://www.ti.com/lit/ds/symlink/ne5532.pdf>
- [16] SLOS209G, *TLV277x, TLV277xA – Family of 2.7-V high-slew-rate rail-to-rail output operation amplifiers with shutdown*, styczeń 1998 – luty 2004
<http://www.ti.com/lit/ds/symlink/tlv2774.pdf>
- [17] SNVS090E, *LM341/LM78MXX Series 3-Terminal Positive Voltage Regulators*, maj 2004 – sierpień 2005
<http://www.ti.com/lit/ds/symlink/lm341.pdf>
- [18] UM1472 User manual, *Discovery kit for STM32F407/417 lines*
<http://www.st.com/stm32>
- [19] *Arduino Nano (v2.3) User Manual*
<https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>
- [20] Arduino Software
<https://www.arduino.cc/en/Main/Software>
- [21] castAR
<http://technicalillusions.com/>
- [22] Estimote Beacon
<http://developer.estimote.com/>
- [23] Galileo
http://www.esa.int/Our_Activities/Navigation/The_future_-_Galileo/What_is_Galileo
- [24] Global Positioning System
<http://www.gps.gov/>
- [25] GLONASS
<https://www.glonass-iac.ru/en/>

- [26] Kąty Eulera
https://pl.wikipedia.org/wiki/K%C4%85ty_Eulera
- [27] Oculus Rift Development Kit 2
<https://vrwiki.wikispaces.com/Oculus+Rift+Development+Kit+2>
- [28] Power Glove
https://en.wikipedia.org/wiki/Power_Glove
- [29] *Reach Out And Touch Your Data*, IBM, BYTE, lipiec 1990
<http://research.microsoft.com/en-us/um/people/bibuxton/buxtoncollection/a/pdf/BYTE%20Glove%20Comparision.pdf>
- [30] *ULTRASONIC SENSOR (GENERAL), 40ST-12*
https://www.maritex.com.pl/media/uploads/PRODUKTY_PDF/sens/40str-12.pdf
- [31] The Pi 3D scanner project
<http://www.pi3dscan.com/>