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 Marcina Peczarskiego

Czerwiec 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 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 wykorzystując do tego celu ultradźwięki o częstotliwości 40 kHz. Zastosowana metoda umożliwia pomiar z wysoką rozdzielczością dochodzącą do 0,5 mm.

Słowa kluczowe

położenie, orientacja, 3D, ultradźwięki

Dziedzina pracy (kody wg programu Socrates-Erasmus)

11.3 Informatyka

Klasyfikacja tematyczna

TODO: dodać

Tytuł pracy w języku angielskim

Determining position and orientation in space using ultrasound

Spis treści

Wprowadzenie

W dzisiejszym świecie obserwujemy coraz większe zapotrzebowanie na urządzenia, które potrafią określić swoje położenie jak i orientację w otaczającej je przestrzeni. Urządzenia takie mają szerokie zastosowanie w wielu dziedzinach m.in. w wirtualnej rzeczywistości, rozszerzonej rzeczywistości, podczas skanowania trójwymiarowego czy kartografii. Przykładowo okulary do wirtualnej rzeczywistości takie jak *Oculus Rift* [14] czy *castAR* [16], muszą uwzględnić położenie jaki i orientację głowy by na tej podstawie wyświetlić użytkownikowi odpowiednią treść.

Z biegiem lat powstało wiele rozwiązań tego problemu, do najczęściej stosowanych możemy zaliczyć:

1. wykorzystanie akcelerometrów, żyroskopów i magnetometrów - takie rozwiązanie zastosowano w *Oculus Rift development kit*, zaletą metody jest stosunkowo prosta konstrukcja jak 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 zachowanie się głowy, jednak rozwiązanie to jest dalekie od idealnego o czym może świadczyć fakt, że w kolejnej wersji urządzenia dodano śledzenie głowy przez zewnętrzną kamerę.

2. projektowanie światła strukturalnego na otoczenie i zbieranie informacji o strukturze światła odbitego za pomocą sensorów, zazwyczaj kamer - taką metodę wykorzystano w *Microsoft Kinnect* [17], urządzenie projektuje na otoczenie stały wzór punktów, następnie kamerą na podczerwień zbierana jest informacja o zniekształceniu danego wzoru i na tej podstawie odtwarzana jest trójwymiarowa struktura otoczenia jak i położenie urządzenia w tym otoczeniu. Podobną metodę wykorzystuje *Oculus Rift development kit 2* [15] jak i *castAR* [16], tutaj za źródła światła służą diody podczerwone umieszczone na okularach, światło przez nie emitowane jest rejestrowane poprzez kamerę umieszczoną przed użytkownikiem. Komputer na podstawie względnego położenia widocznych punktów określa położenie i orientację okularów w przestrzeni. Zaletą tego rozwiązania są stałe punkty odniesienia (kamera) jak i możliwość pomiaru wielu puntów na raz. Do wad należy zaliczyć stosunkowo niską rozdzielczość szczególnie w osi Z jak i duży strumień danych do obróbki.

3. wykorzystanie wielu zdjęć zawierających stałe (nie zmieniające się w czasie) obiekty, na postawie których wyznaczana jest pozycja kamery względem nich lub odwrotnie, kamera (lub wiele kamer) jest punktem stałym, a wyznaczana jest pozycja fotografowanych obiektów - taką metodę wykorzystano w *VidialSFM* [18], jak i w *The Pi 3D scanner project* [20], rozwiązanie to cechuje się również dość niską rozdzielczością.

W niniejszej pracy przedstawiono prototyp oparty na zmodyfikowanej metodzie drugiej, który zamiast światła wykorzystuje ultradźwięki. Podejście to zapewnia dużo większą dokładność szczególnie w osi Z, prostotę budowy jak i dużo niższą cenę. Urządzenie składa się z

dwóch części: odbiornika, na którym umieszczone są trzy mikrofony, i nadajnika, na którym znajdują się cztery głośniki. Pomiędzy nimi dokonywany jest pomiar odległości z rozdzielczością dochodzącą do 0,5 mm dzięki czemu prototyp jest w stanie określić położenie nadajnika w przestrzeni jak i jego orientację. Pomiar odległości dokonywany jest za pomocą ultradźwięków o częstotliwości 40 kHz, mimo iż metoda jest znana od wielu lat, to z uwagi na relatywnie dużą długość fali ultradźwiękowej, która dla częstotliwości 40 kHz wynosi około 8 mm jest do precyzyjnych pomiarów rzadko stosowana. W niniejszej pracy udało się to pozorne ograniczenie przezwyciężyć. Ostatecznie urządzenie jest w stanie śledzić położenie nadajnika z rozdzielczością 5000 px × 5000 px × 5000 px na przestrzeni sześcianu o rozmiarach 2,5 m × 2,5 m × 2,5 m przy czasie odświeżania rzędu 350 ms. Warto podkreślić, że wielkość sześcianu została ograniczona ze względów praktycznych i bez większego problemu można ją zwiększyć zachowując stosunek rozdzielczości na metr. Jeśli chodzi o stosunkowo długi czas odświeżania to głównym czynnikiem na niego wpływającym jest czas jaki potrzebuje fala dźwiękowa by się rozproszyć, tak by jej odbicia od powierzchni ścian nie wpływały na kolejne pomiary.

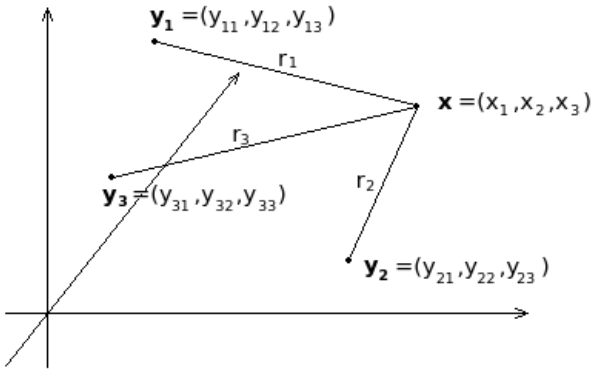
Należy również wspomnieć o innych pracach, które w podobny sposób podchodzą do problemu pozycjonowania, przykładowo w pracy *Ultrasonic 3D Wireless Computer Mouse* [19] przedstawiono prototyp myszki 3D, której pozycja w przestrzeni wyznaczana jest za pomocą ultradźwięków, niemniej wykorzystany przez autorów algorytm wyznaczania odległości jest dużo bardziej podatny na błędy. Dodatkowo praca koncentruje się jedynie na określaniu położenia myszki w przestrzeni, bez wyznaczania jej orientacji. Kolejnym ciekawym rozwiązaniem jest rękawica dla graczy *Power Glove* [21] [22] dla *Nintendo Entertainment System* wydana w roku 1989, ona również wykorzystuje ultradźwięki do wyznaczania pozycji rękawicy w przestrzeni, jednak precyzja urządzenia jest wyjątkowo niska przez co rękawica nie odniosła znaczącego sukcesu komercyjnego.

Rozdział 1

Podstawy teoretyczne

1.1. Wyznaczanie położenia na podstawie odległości od punktów stałych

Zadaniem prezentowanego prototypu jest wyznaczenie położenia nadajnika na podstawie podległości od znanych punktów stałych. Niech $\boldsymbol{x}$ będzie szukanym punktem w przestrzeni, $\boldsymbol{y_1}, \boldsymbol{y_2}, \boldsymbol{y_3}$ stałymi punktami o znanym położeniu, a r_1, r_2, r_3 wyznaczonymi odległościami, rysunek 1.1.



Rysunek 1.1: Wyznaczenie położenia na podstawie odległości od punktów stałych

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

$$\begin{cases} |\boldsymbol{y_1} - \boldsymbol{x}| = r_1 \\ |\boldsymbol{y_2} - \boldsymbol{x}| = r_2 \\ |\boldsymbol{y_3} - \boldsymbol{x}| = r_3 \end{cases}$$

co dla przestrzeni euklidesowej daje:

$$\begin{cases} (y_{11} - x_1)^2 + (y_{12} - x_2)^2 + (y_{13} - x_3)^2 = r_1^2 \\ (y_{21} - x_1)^2 + (y_{22} - x_2)^2 + (y_{23} - x_3)^2 = r_2^2 \\ (y_{31} - x_1)^2 + (y_{32} - x_2)^2 + (y_{33} - x_3)^2 = r_3^2 \end{cases}$$

Zauważmy, że znając odległość $\boldsymbol{x}$ od jednego punktu stałego wiemy, że $\boldsymbol{x}$ będzie leżał na sferze o promieniu równym danej odległości. Znając dwie odległości możemy ograniczyć

poszukiwania do części wspólnej dwóch sfer, czyli okręgu. Dla znanych trzech odległości dostajemy część wspólną okręgu ze sferą, czyli dwa możliwe punkty. Mimo, iż trzy odległości nie są wystarczające do jednoznacznego wyznaczenia $\boldsymbol{x}$, to otrzymane rozwiązania są na ogół od siebie daleko oddalone i bez większego problemu możemy założyć, że jedno z rozwiązań jest naszym szukanym punktem.

Ponieważ mamy pełną dowolność w doborze punktów stałych $\boldsymbol{y_i}$ dla uproszenia możemy przyjąć: $\boldsymbol{y_1} = (\frac{a}{2}, 0, 0), \boldsymbol{y_2} = (-\frac{a}{2}, 0, 0), \boldsymbol{y_3} = (0, h, 0)$, czyli $\boldsymbol{y_1}, \boldsymbol{y_2}, \boldsymbol{y_3}$ są wierzchołkami trójkąta równoramiennego o wysokości h i podstawie a, wtedy układ równań upraszcza się do postaci:

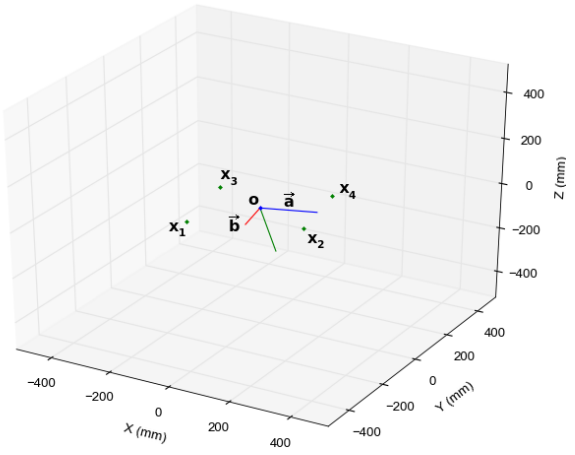
$$\begin{cases} (\frac{a}{2} - x_1)^2 + x_2^2 + x_3^2 = r_1^2 \\ (\frac{a}{2} + x_1)^2 + x_2^2 + x_3^2 = r_2^2 \\ x_1^2 + (h - x_2)^2 + x_3^2 = r_3^2 \end{cases}$$

z czego ostatecznie otrzymujemy:

$$\begin{cases} x_1 = \frac{r_2^2 - r_1^2}{2a} \\ x_2 = \frac{6r_2^2 - 4r_3^2 - 2r_1^2 + a^2}{8h} + \frac{h}{2} \\ x_3 = \pm\sqrt{r_3^2 - x_1^2 - (h - x_2)^2} \end{cases}$$

1.2. Wyznaczanie orientacji w przestrzeni

Kolejną rzeczą jaka nas interesuje jest orientacja w przestrzeni nadajnika. Standardowo orientację zdefiniuje się jako kąty Eulera [23]: przechylenia, pochylenia oraz odchylenia wokół osi związanych z obiektem, jednak na nasze potrzeby wygodniej posługiwać się dwoma prostopadłymi wektorami, które jednoznacznie wyznaczają ów kąty. Prezentowany prototyp wyznacza położenie czterech punktów $\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \boldsymbol{x_4}$ znajdujących się na nadajniku, a następnie na ich podstawie wyznacza poszukiwane dwa prostopadłe wektory które jednoznacznie definiują jego orientację, rysunek 1.2.



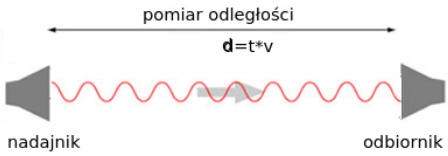
Rysunek 1.2: Wyznaczanie orientacji na podstawie czterech punktów

Na rysunku punkt $\boldsymbol{o} = (\boldsymbol{x_1} + \boldsymbol{x_2} + \boldsymbol{x_3} + \boldsymbol{x_4})/4$ jest szukanym położeniem nadajnika, a wektory: $\overrightarrow{\boldsymbol{a}} = (\boldsymbol{x_2} + \boldsymbol{x_4})/2 - \boldsymbol{o}$ i $\overrightarrow{\boldsymbol{b}} = (\boldsymbol{x_1} + \boldsymbol{x_2})/2 - \boldsymbol{o}$ 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 potrzebujemy jedynie trzech kątów, w sumie daje to sześć niewiadomych. W prezentowanym podejściu wyznaczanych jest dwanaście niewiadomych co daje dwukrotną nadmiarowość danych. Jest to efekt pożądany, ów nadmiarowość wykorzystywana jest do korekcji błędów pomiarowych.

1.3. Pomiar odległości za pomocą ultradźwięków

Prezentowane urządzenie do pomiaru odległości wykorzystuje metodę pomiaru czasu jaki 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 powietrzy oraz czas jaki fala dźwiękowa potrzebowała aby pokonać dany dystans jesteśmy w stanie wyznaczyć szukaną odległość.

Prędkość dźwięku w powietrzu jest mocno zależy od panujących warunków atmosferycznych [1], głównym czynnikiem na nią wpływającym jest temperatura. Dla gazu doskonałego prędkość wyraża się wzorem:

$$V_{air} = 331,3\frac{m}{s}\sqrt{1+\frac{T}{273,15}}$$

gdzie T jest temperaturą w stopniach Celsjusza (°C). Wzór ten możemy przybliżyć za pomocą rozwinięcia Taylora dla temperatur bliskich temperaturze pokojowej 25 °C:

$$V_{air} = (346,13 + 0,580(T - 25\,^{\circ}))\frac{m}{s}$$

Mimo że współczynnik temperaturowy jest stosunkowo mały i stanowi jedynie 0,17 % całej prędkości to przy pomiarach odległości rzędu metrów i rozdzielczościach rzędu milimetrów staje się on istotny, dlatego w prezentowanym prototypie istotną częścią stanowi kalibracja wstępna, podczas której wyznaczana jest aktualna prędkość dźwięku.

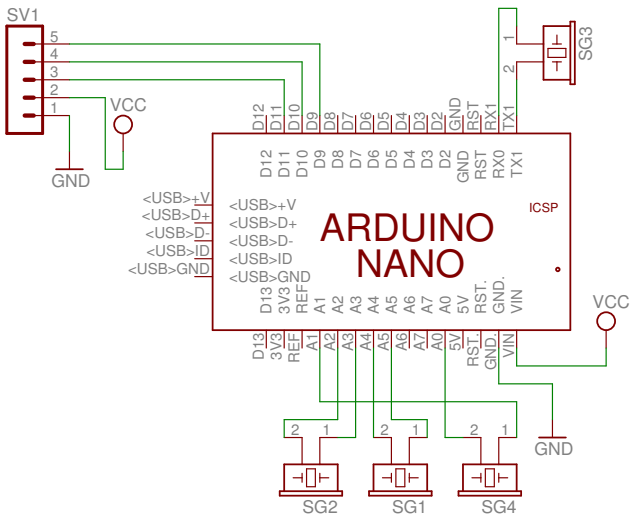
Rozdział 2

Nadajnik ultradźwiękowy

Prezentowany prototyp składa się z dwóch części: nadajnika i odbiornika pomiędzy którymi dokonywany jest pomiar odległości. W niniejszym rozdziale opisana została budowa nadajnika.

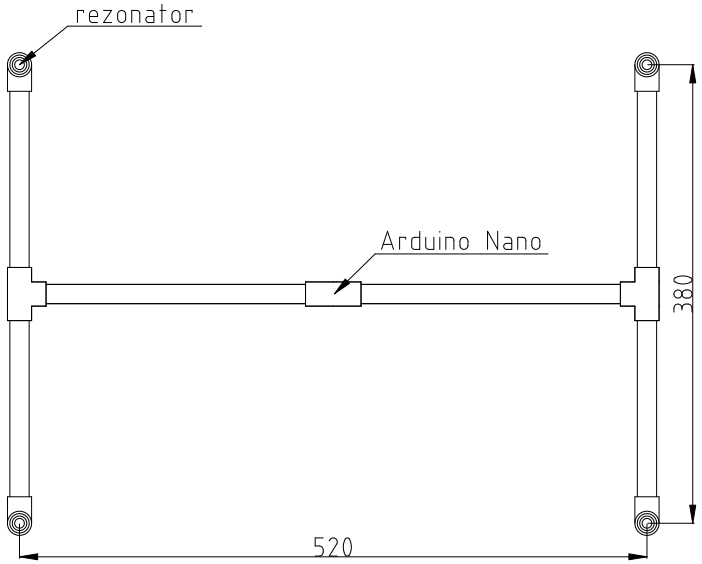
2.1. Budowa i zasada działania

Nadajnik zbudowany został w oparciu o płytkę prototypową *Arduino Nano* [4], która składa się z procesora ATmega328 [12] taktowanego 16 MHz rezonatorem kwarcowym, 5 V stabilizatora napięcia oraz układ FT232RL umożliwiający jej programowanie poprzez port USB ze środowiska *Arduino* [13]. Do *Arduino Nano* połączono bezpośrednio cztery nadajniki ultradźwiękowe (głośniki, rezonatory piezoelektryczne) typu 40ST-12 [5]. Schemat połączeń przedstawiono na rysunku 2.1.



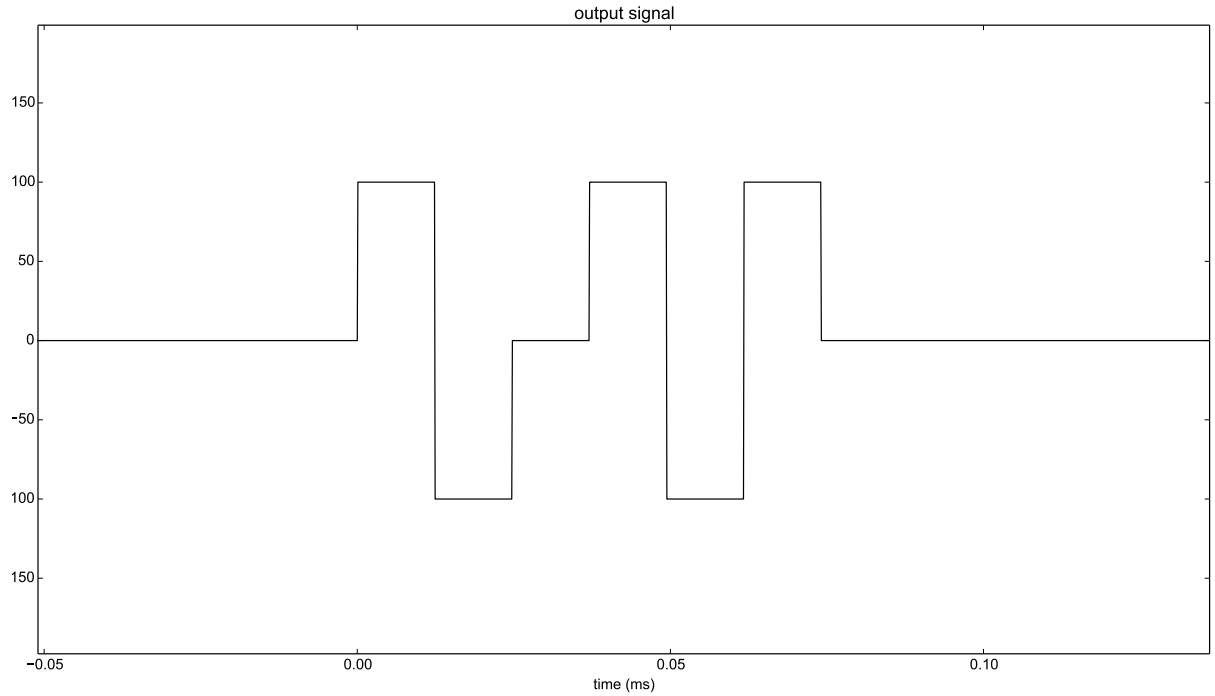
Rysunek 2.1: Schemat nadajnika.

Całość umieszczona została na ramie w kształcie litery H wykonanej z rurek PCV. Rezonatory zostały dodatkowo odizolowane od ramy rzepami co ułatwia ich demontaż jak i skutecznie zapobiega przenoszeniu się drgań.



Rysunek 2.2: Szkic ramy nadajnika

Android Nano połączony jest z odbiornikiem 6 m kablem, którym przesyłane są sygnały sterujące jak i zasilanie. Do sterowania wykorzystywane są trzy przewody, dwa z nich informują który z głośników ma w danym momencie nadawać, trzeci służy jako wyzwalacz. Na potrzeby nadajnika powstało oprogramowanie dla procesora ATmega328 napisane w C generujące nadawany sygnał. Cała logika programu mieści się w obsłudze przerwania sprzętowego, które reaguje na zmianą stanu logicznego na wyzwalaczu. Po uruchomieniu przerwania oprogramowanie wysyła z góry zdefiniowany sygnał na odpowiedni rezonator. Nadawany sygnał jest tak dobrany by dało się go w prosty sposób wyodrębnić i składa się w dwóch części: część wzbudzającej oraz części tłumiącej. Okres impulsów sygnału jest równy częstotliwością rezonansową przetworników, dodatkowo część tłumiąca jest przesunięty względem części wyzwalającej o 180 stopni. Rysunek 2.3 przedstawia sygnał jakim wysterowany jest przetwornik piezoelektryczny.



Rysunek 2.3: Sygnał wysterowania przetwornika piezoelektrycznego

2.2. Dobór rezonatorów piezoelektrycznych

Głównym problemem podczas konstrukcji nadajnika okazał się dobór odpowiednich rezonatorów piezoelektrycznych. Mimo, że producent wykorzystanych rezonatorów piezoelektrycznych zapewnia ich pracę zakresie: 40 kHz ± 1 kHz, to taki rozrzut okazał się zbyt duży, dlatego z 30 rezonatorów (15 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.

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

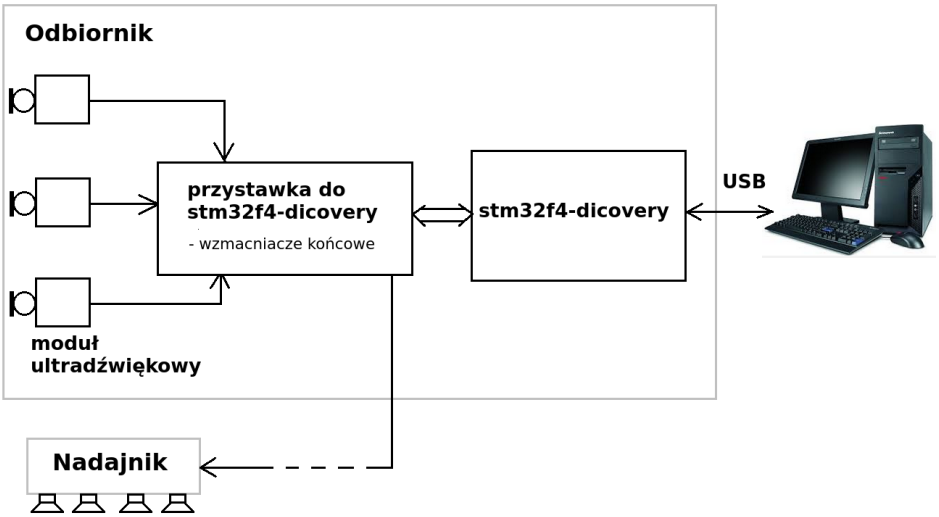
Tabela 2.1: Częstotliwości rezonansowe przetworników piezoelektrycznych

Rozdział 3

Odbiornik

Centralną częścią prezentowanego prototypu jest odbiornik. Jego zadaniem jest wysyłanie sygnałów sterujących do nadajnika, zbieranie ultradźwięków z otoczenia oraz przesłanie ich do dalszej analizy do komputera. Składa się on z sześciu części (rysunek 3.1):

- trzech modułów ultradźwiękowych przetwarzających dźwięk na sygnał elektryczny
- płytki prototypowej *stm32f4-discovery* [6] odpowiadającej za komunikację z komputerem
- przystawki do *stm32f4-discovery* przystosowującej sygnały elektryczne z modułów ultradźwiękowych do poziomów akceptowalnych przez *stm32f4-discovery*
- ramy, na której umieszczone są moduły ultradźwiękowe



Rysunek 3.1: Szkic odbiornika

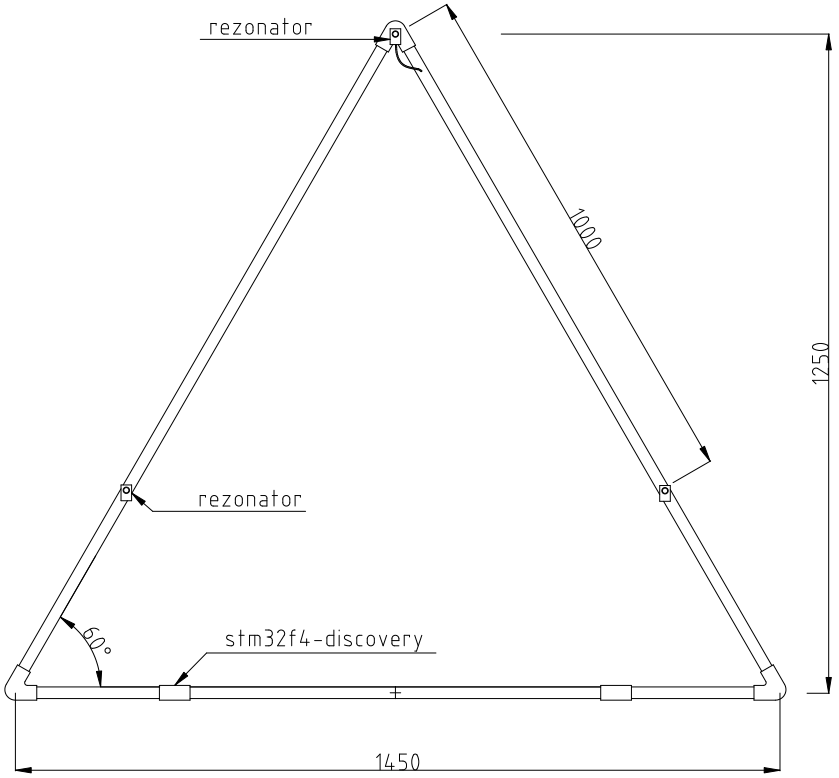
3.1. Budowa i zasada działania

Głównym elementem odbiornika stanowi płytka prototypowa *stm32f4-discovery* [6], jej zadaniem jest umożliwienie komunikacji wszystkich komponentów z komputerem. *Stm32f4-discovery* oparta jest na procesorze STM32F407VGT6 [7] jest to procesor typu ARM Cortex

M4, który posiada wbudowane trzy 12 bitowe przetworniki analogowo-cyfrowe umożliwiające próbkowanie z prędkością dochodzącą do 2,4 MSPS, przetworniki wykorzystane zostały do próbkowania sygnałów pochodzących z modułów ultradźwiękowych. Procesor umożliwia również komunikację z komputerem poprzez port USB z prędkością 12 MB/s. Płytka prototypowa wyposażona została również w programator/debugger umożliwiający programowanie procesora i jego testowanie poprzez dodatkowy port USB.

Na potrzeby odbiornika powstało dedykowane oprogramowanie napisane w C sterujące procesorem. Oprogramowanie oparte jest na bibliotece *stm32 usb 101* [2] zapewniającej komunikację z komputerem, do której dodana została obsługa przetworników *ADC*. Program poprzez port USB dostaje komunikat, który z czterech nadajników ma nadawać, ta informacja przekazywana jest dalej do nadajnika w raz z sygnałem wyzwalającym, następnie uruchamiane są równocześnie trzy przetworniki *ADC*, które samplują odbierany dźwięk i poprzez DMA zapisują trzy kanały w pamięci procesora. Częstotliwość pracy przetworników ustawiona została na 1,6 MSPS co daje średnio 40 próbek na jeden okres 40 kHz sygnału. Program zapamiętuje 16 kS na każdy kanał, co przy prędkości dźwięku 340 m/s daje maksymalną mierzoną odległość rzędu 2,5 metrów. Po zebraniu w sumie 48 kS, całość przesyłana jest do komputera w celu dalszej analizy. Proces powtarzany jest dla każdego z czterech głośników nadajnika, co w sumie daje 12 sygnałów, na podstawie których wyznaczona zostaje pozycja w przestrzeni oraz orientacja nadajnika.

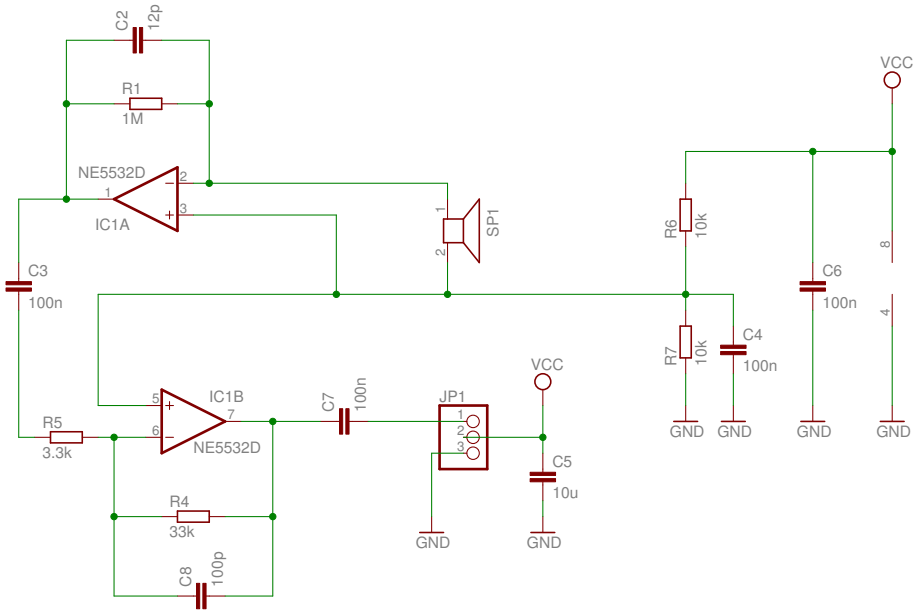
Cała elektronika osadzona została na ramie zbudowanej z rur PCV w kształcie trójkąta (rysunek 3.2), odległości pomiędzy modułami ultradźwiękowymi są z góry znane, co ułatwia dalsze obliczenia.



Rysunek 3.2: Szkic ramy odbiornika

3.2. Budowa modułu ultradźwiękowego

Odbiornik wyposażony został w trzy moduły ultradźwiękowe, których zadaniem jest zbieranie ultradźwięków z trzech oddalonych od siebie punktów. Każdy z modułów wyposażony jest w przetwornik piezoelektryczny (mikrofon) 40SR-12 [5], który przetwarza sygnał akustyczny na odpowiadający mu sygnał elektryczny oraz wzmacniacze operacyjne wstępnie wzmacniające sygnał przesyłany dalej do przystawki *Stm32f4-discovery*. Rysunek 3.3 przedstawia schemat modułu ultradźwiękowego.



Rysunek 3.3: Schemat modułu ultradźwiękowego

Zasada działania: wzmacniacz operacyjny IC1A wraz z kondensatorem C2 i rezystorem R1 pracuje jako przedwzmacniacz ładunkowy [8], ładunek wytworzony na przetworniku piezoelektrycznym SP1 zostaje w całości przeniesiony na kondensator C2 (wzmacniacz stara się utrzymać różnicę potencjałów miedzy dodatnim a ujemnym wejściem na zerowym poziomie), co jest równoważne z pojawieniem się napięcia na kondensatorze zgodnie z równaniem $U = \frac{q}{C}$. Rezystor R1 ustawia napięcia spoczynkowe układu na poziome $\frac{1}{2}$ Vcc jak i rozładowuje kondensator C2. R1 i C2 działają również jako filtr górnoprzepustowy.

Wzmacniacz IC1B wraz z rezystorami R5 i R4 pracuje jako zwykły wzmacniacz napięciowy wzbogacony o filtr górno (kondensatory C3, C7) i dolnoprzepustowy (kondensator C8).

Aby zminimalizować zakłócenia całość oparta jest na niskoszumowych wzmacniaczach operacyjnych mieszczących się w jednym układzie scalonym NE5532 [9], dodatkowo płytka drukowana jest ekranowana.

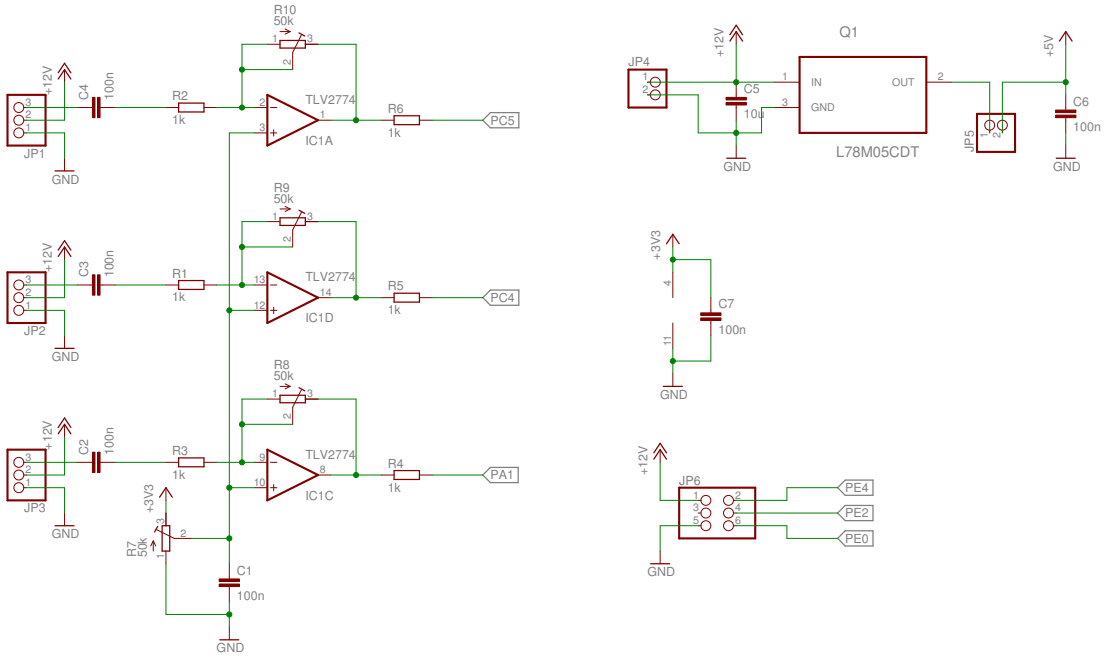
Wzmocniony sygnał poprzez wtyczkę JP1 doprowadzony jest do przystawki współpracującej z *stm32f4-discovery*.

3.3. Przystawka do *stm32f4-discovery*

Sygnał z modułów ultradźwiękowych doprowadzany jest do *stm32f4-discovery* poprzez specjalną przystawkę. Schemat przystawki przedstawiony jest na rysunku 3.4.

Jej zadaniem jest przystosowanie maksymalnych amplitud zebranych sygnałów do wartości akceptowalnych przez przetworniki analogowo-cyfrowe procesora STM32F407VGT6, które muszą mieścić się w zakresie od 0 V do 3,3 V.

Do tego celu zastosowany został układ TLV2774 [10], który zawiera w sobie 4 wzmacniacze operacyjne typu rail-to-rail, trzy z nich wykorzystane zostały jako ostatni stopień wzmocnienia sygnałów ultradźwiękowych. Wzmacniacze operacyjne pracują w układzie odwracającym z nastawianym wzmocnieniem (potencjometry R8, R9, R10), do których podłączone zostało wspólne regulowane napięcie odniesienia (potencjometr R7). Przystawka zawiera również stabilizator napięcia LM78M05CDT [11], który po podłączeniu 12 V baterii do JP4, dostarcza napięcie zasilające do wszystkich komponentów. Istnieje możliwość odcięcia zasilania poprzez rozwarcie zworkę JP5, co jest konieczne podczas programowania STM32F407VGT6. Z przystawki wyprowadzono również sygnały sterujące nadajnikiem wraz z zasilaniem (wtyczka JP6).



Rysunek 3.4: Schemat przystawki do *stm32f4-discovery*

Rozdział 4

Przetwarzanie danych zebranych z odbiorników

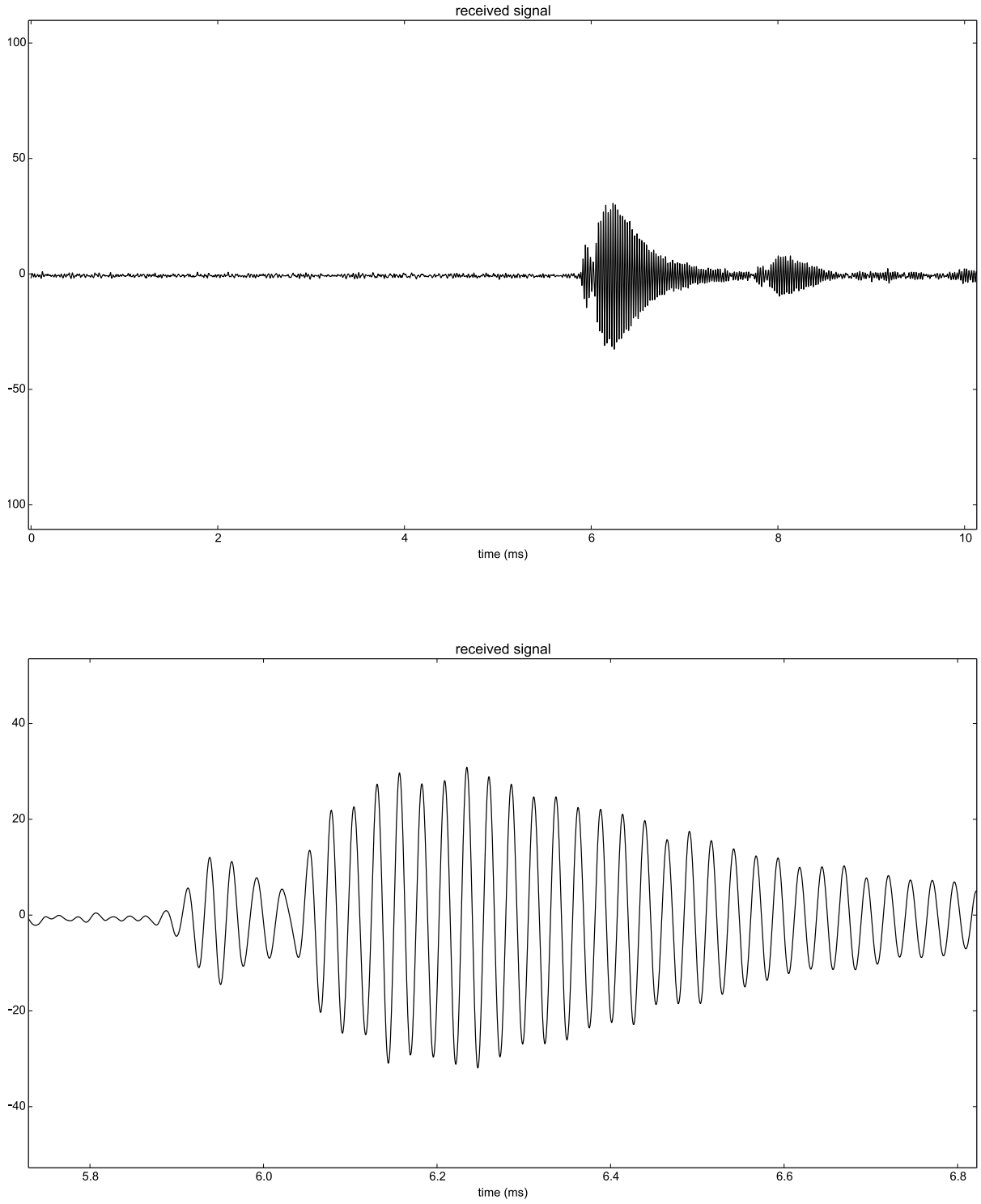
W ramach pracy powstało oprogramowanie uruchamiane na komputerze, którego zadaniem jest przekształcenie surowych danych ultradźwiękowych z odbiornika na położenie nadajnika w przestrzeni oraz jego orientację. Oprogramowanie podzielone zostało na trzy części: bibliotekę *mp3d* zajmującą się sterowaniem odbiornika, 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 stanowi biblioteka *mp3d*, została ona podzielona na pięć modułów:

1. *com.py* - moduł odpowiedzialny za komunikację z odbiornikiem
2. *find_pattern.py* - moduł odpowiedzialny za wyznaczanie odległości poprzez wyszukanie wzorca w odebranym sygnale
3. *xyz.py* - moduł odpowiedzialny za wyznaczenie pozycji i orientacji nadajnika oraz za weryfikację, czy zebrane dane odpowiadają modelowanej rzeczywistości
4. *info.py* - moduł wyświetlający informację o sile odbieranego sygnału
5. *ply.py* - moduł odpowiedzialny za eksportowanie danych do formaty *.ply* (Polygon File Format) obsługiwanego przez większość programów do obróbki grafiki 3D.

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

Za komunikację z odbiornikiem odpowiedzialny jest moduł *com.py*, pracuje on w oddzielnym wątku, w którym cyklicznie wysyłane są żądania by dany głośnik nadał sygnał, następnie odbierane są sygnały z trzech mikrofonów. Ta czynność powtarzana jest dla każdego z czterech głośników nadajnika. Zebrane dwanaście sygnałów po wstępnej filtracji przekazywane są dalej do *find_pattern.py*. Cały cykl powtarzany jest co 350 ms. Na rysunku 4.1 przedstawiono odebrany sygnał z jednego mikrofonu.



Rysunek 4.1: Sygnał odebrany przez moduł *com.py*. Odległość między nadajnikiem a odbiornikiem wynosi 2 metry. Na górnym wykresie po prawej stronie widoczne są również sygnały odbite od ścian.

4.2. Wyznaczanie odległości, moduł *find_pattern.py*

Moduł *find_pattern.py* odpowiada za obliczenie odległości pomiędzy czterema głośnikami i trzema mikrofonami. Odległość wyznaczana jest poprzez wyszukanie położenia *wzorca* odpowiadającego czołu nadanego sygnału w każdym z 12 odebranych sygnałów (rysunek 4.1). *Wzorzec* początkowo wprowadzany jest podczas kalibracji, następnie jest ciągle aktualizowany (podmieniany z sygnałem pasującym do *wzorca*).

Do wyszukania *wzorca* wewnątrz sygnału wykorzystana jest metoda najmniejszego błędu średniokwadratowego:

niech $w(t)$ dla $t = 0..n-1$ będzie szukanym *wzorcem*, a $f(x)$ odebranym sygnałem, wtedy możemy znaleźć takie a, że błąd średniokwadratowy $E(x)$ pomiędzy $w(t)$ i $af(t+x)$ jest minimalny, możemy to zapisać w postaci:

$$E(x) = \min_{a \in R}\{\sum_{t=0}^{n-1}(w(t) - af(t+x))^2\}$$

po podniesieniu wyrażenia w nawiasie do kwadratu otrzymujemy:

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

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

aby wyliczyć $E(x)$ należy zminimalizować wyrażenie:

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

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

$$a = \frac{\sum\limits_{t=0}^{n-1} w(t)f(t+x)}{\sum\limits_{t=0}^{n-1} f^2(t+x)}$$

ostatecznie dostajemy wzór na $E(x)$:

$$E(x) = \sum_{t=0}^{n-1} w^2(t) - \frac{(\sum\limits_{t=0}^{n-1} w(t)f(t+x))^2}{\sum\limits_{t=0}^{n-1} f^2(t+x)}$$

Zauważmy, że dzięki skalowaniu sygnału $f(x)$ skalarem a zamiast *wzorcowa* $w(x)$ otrzymany błąd $E(x)$ nie zależy od siły odebranego sygnału co ułatwia porównanie błędów w dwóch różnych miejscach, zależy on jednak od siły sygnału wzorcowego, aby się od niego uniezależnić możemy wyznaczyć błąd względny $e(x)$:

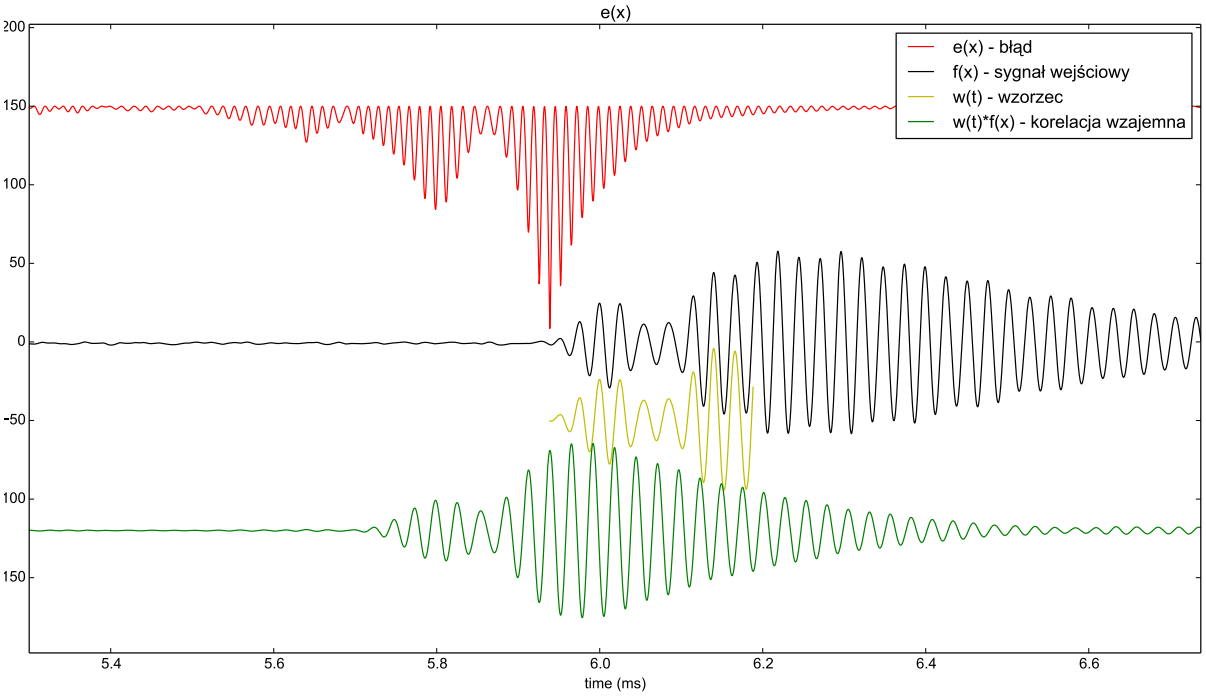
$$e(x) = \frac{E(x)}{\sum\limits_{t=0}^{n-1} w^2(t)}$$

po podstawieniu $E(x)$ dostajemy:

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

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

Wynikiem obliczeń modułu *find_pattern.py* jest cała funkcja $e(x)$, na podstawie której moduł *xyz.py* wyznaczy pozycję głowicy w przestrzeni jak i jego orientację uwzględniając przy tym kształt głowicy (nadmiarowość danych) jak i prawdopodobieństwo że znaleziony wzorzec jest na danej pozycji. Na rysunku 4.2 przedstawiono wynik przetwarzania sygnału poprzez moduł *find_pattern.py*.



Rysunek 4.2: Sygnał przetworzony przez moduł *find_pattern.py*.

Analiza złożoności obliczeniowej: wyliczenie $\sum_{t=0}^{n-1} w^2(t)$ jak i $\sum_{t=0}^{n-1} f^2(t+x)$ wymaga jedynie liniowej liczby operacji, a $\sum_{t=0}^{n-1} w(t)f(t+x)$ jest korelacją wzajemną funkcji $w(t)$ i $f(x)$, którą można wyliczyć w czasie $m\log(m)$ korzystając z szybkiej transformacji Fouriera [3], gdzie $m = \max\{|w|, |f|\}$, czyli wyliczenie całej funkcji $e(x)$ wymaga jedynie $m\log(m)$ operacji.

Kolejnym zadaniem modułu *find_pattern.py* jest uaktualnianie *wzorca*. Wraz ze zmianą kąta nachylenia nadajnika względem odbiornika zmienia się kształt odbieranego sygnału, dlatego jeśli badany sygnał zawiera szukany *wzorzec* oraz spełnione są następujące warunki:

- moc sygnału pasującego do *wzorca* jest wystarczająco duża

- błąd względny pomiędzy sygnałem a *wzorcem* jest mały
- moduł *xyz.py* "uzna", że otrzymane pozycje pasują do modelowanej rzeczywistości

to moduł uaktualni *wzorzec*.

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

Ostatnim etapem obliczeń jest wyznaczenie współrzędnych kartezjańskich czterech głośników znajdujących się na nadajniku, na podstawie których obliczana jest pozycja jaki i orientacja całego nadajnika. Tym etapem zajmuje się moduł *xyz.py*. Moduł zawiera również prosty model rzeczywistości, którego celem jest wyłapanie niepoprawnych danych o położeniu. Niepoprawne dane powstają gdy część głośników traci widoczność z mikrofonami np. gdy zostaną przysłonięte. W ramach modelowanej rzeczywistości wyliczane są:

- prędkość z jaką porusza się każdy z głośników
- kształt nadajnika - położenie głośników względem siebie w przestrzeni

Jeśli któreś z wyliczeń odbiega od z góry ustalonej normy to dany pomiar uznawany jest za nieprawidłowy.

Współrzędne głośników wyznaczane są na podstawie dwunastu funkcji $e_{ij}(x)$ dla $i = 1, ..., 4$ i $j = 1, ..., 3$, które zawierające informację o odległości pomiędzy głośnikiem i oraz mikrofonem j. Dla każdej funkcji $e_{ij}(x)$ wyszukiwane są pozycje minimów lokalnych tych funkcji, z których następnie wybierane jest t pozycji $x_1^{ij}, ..., x_t^{ij}$ o najmniejszych wartościach, gdzie t jest stałą wewnętrzną modułu *xyz.py*. Pozycje te są odległościami, dla których błąd średniokwadratowy $e_{ij}(x_k^{ij})$ dla $k = 1, ..., t$ pomiędzy szukanym sygnałem (*wzorcem*), a odebranym sygnałem na pozycji x_k^{ij} jest lokalnym minimum jak i jest mały w sensie globalnym, pozycje x_k^{ij} można również interpretować jako najbardziej prawdopodobne odległości pomiędzy głośnikiem, a mikrofonem.

Do wyliczenia położenia głośników w przestrzeni wykorzystywane są wszystkie możliwe kombinacje wyznaczonych odległości, zauważmy że jest ich 12^t. Każda z możliwych kombinacji sprawdzana jest czy jest akceptowalna dla modelowanej rzeczywistości, z tych które są akceptowalne wybierana jest kombinacja o najmniejszym sumarycznym błędzie średniokwadratowym, czyli najbardziej prawdopodobna.

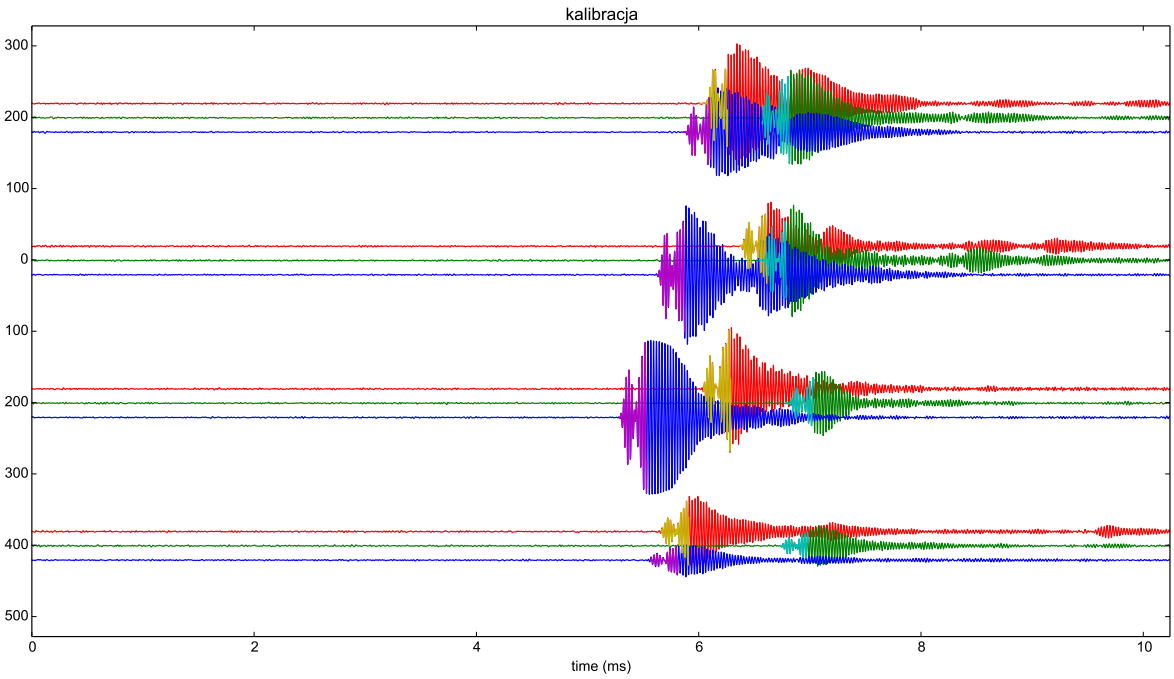
Ostatecznym wynikiem moduły *xyz.py* są wartości x^{ij}, czyli najbardziej prawdopodobne odległości pomiędzy i-tym głośnikiem (gdzie $i = 1, ..., 4$), a j-tym mikrofonem, na podstawie których wyznaczana jest pozycja oraz orientacja nadajnika.

Wartości x^{ij} przekazywane są również z powrotem do moduły *find_pattern.py* w celu aktualizacji *wzorców*.

4.4. Kalibracja

Przed każdym uruchomieniem urządzenia wymagana jest kalibracja, która sprowadza się do ustawienia nadajnika w odległości 2 metrów od odbiornika, uruchomienia programu *save-pattern.py* oraz zaznaczenia 12 obszarów będącym czołem odebranego sygału. Obszary te wykorzystane zostaną jako *wzorce* podczas mierzenia odległości. Program wyliczy również aktualną prędkość dźwięku uwzględniając odległość nadajnika od odbiornika.

Rysunek 4.3 przedstawia widok 12 sygnałów wraz z zaznaczonymi wzorcami.



Rysunek 4.3: Kalibracja, widok 12 sygnałów z zaznaczonymi *wzorcami*

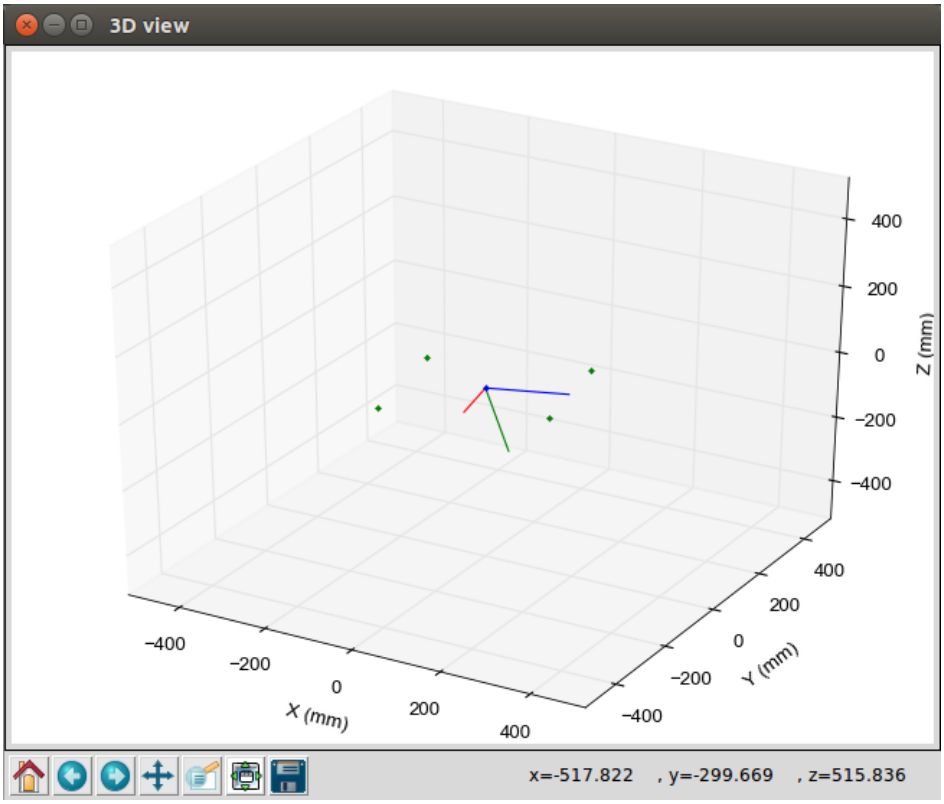
4.5. Obsługa programu *scan.py*

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

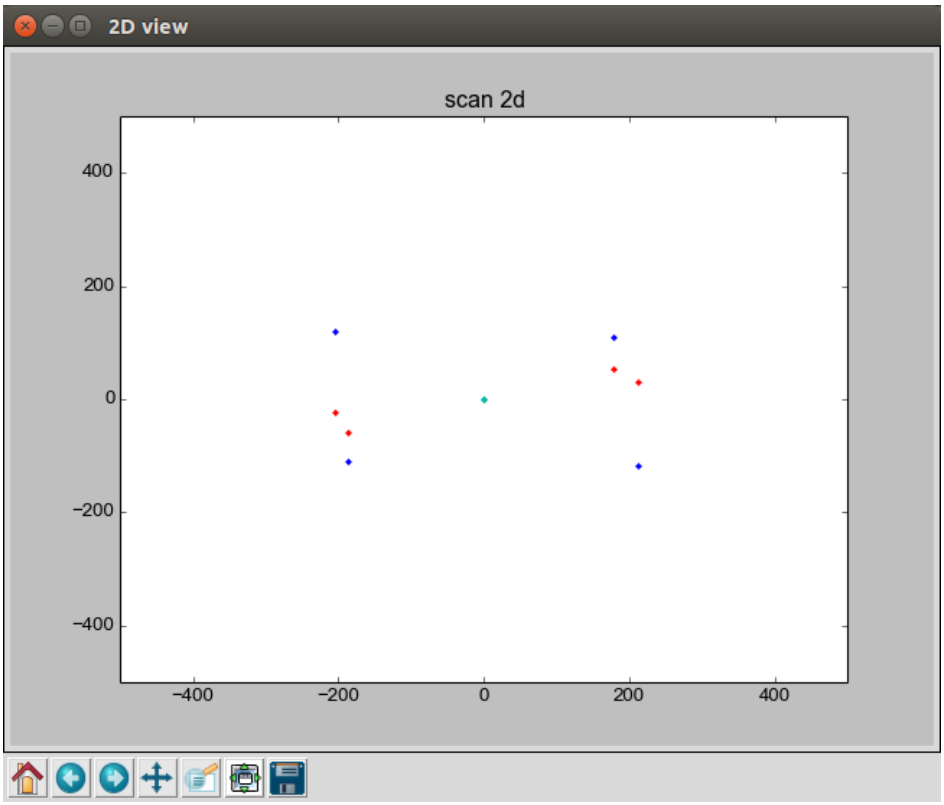
- widok 3D prezentujący położenie nadajnika w przestrzeni. W oknie widoczne są cztery punkty reprezentujące głośniki nadajnika oraz trzy wektory normalne reprezentujące jego orientację jak i położenie, rysunek 4.4
- widok 2D, na którym przedstawiono nałożone na siebie dwa rzuty prostopadłe nadajnika na płaszczyzny XY i XZ, rysunek 4.5
- widok mocy sygnału, który informujący o sile odbieranego sygnału oraz o błędzie pomiędzy *wzorcem* a odebranym sygnałem, rysunek 4.6

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 na widoku mocy sygnału.

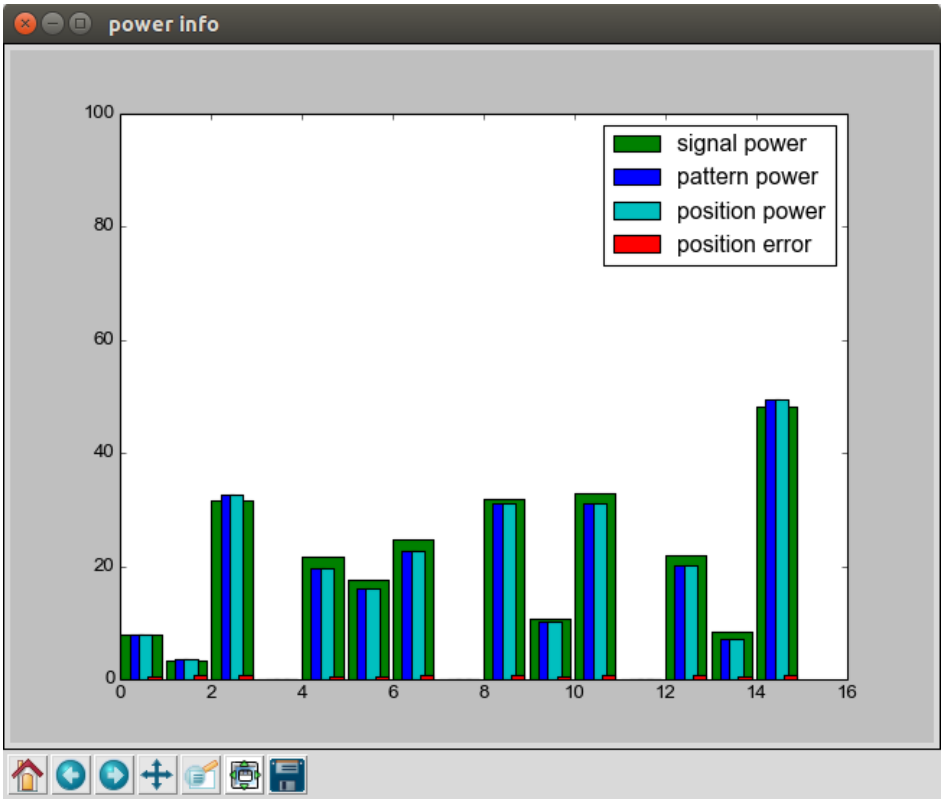
Dla celów demonstracyjnych istnieje możliwość rysowania (stawiania punktów) w przestrzeni. Aby zapamiętać punkty należy wcisnąć klawisz "p", aby zapisać wynik rysowania należy wcisnąć klawisz "s", do wyczyszczenia obrazu służy klawisz "c". Rysunek [TODO] przedstawia przykładowy rysunek w przestrzeni.



Rysunek 4.4: Widok 3D



Rysunek 4.5: Widok 2D



Rysunek 4.6: Widok informujący o mocy sygnału oraz błędzie pomiędzy *wzorcem* a sygnałem

Rozdział 5

Podsumowanie

W niniejszej pracy zademonstrowano urządzenie potrafiące określić swoje położenie jak i orientację w przestrzeni wykorzystują do tego celu ultradźwięki. Mimo że wykorzystano fale dźwiękowe o długość 8 mm, pomiar położenia osiąga rozdzielczości rzędu 0,5 mm co znacząco wyróżnia się na tle innych rozwiązań. W tym rozdziale przedstawiono zalety jaki i wady prezentowanego rozwiązania.

Zalety:

- Wysoka rozdzielczość pomiaru. Zastosowanie dużo wyższych częstotliwości próbkowania od częstotliwości nadawanego sygnału, jak i zastosowane wyszukiwanie wzorca pozwoliło wielokrotnie zwiększyć rozdzielczość pomiaru w stosunku to długości wykorzystanej fali dźwiękowej.

- Praktycznie nieograniczona przestrzeń robocza. Mimo, że w prezentowanym prototypie ograniczono przestrzeń roboczą do kwadratu 2,5 m × 2,5 m × 2,5 m, to jest to jedynie ograniczenie praktyczne i bez większych problemów można ją dowolnie zwiększyć. Trzeba jednak pamiętać, że siła sygnału dźwiękowego maleje kwadratowo wraz z odległością więc przy dużo większych odległościach niezbędne będą nadajniki większej mocy.

- Niski koszt budowy.

Wady:

- Duża czułość na zmiany temperatury. Jest to najpoważniejsza wada prezentowanego rozwiązania. Zauważmy, że zmiana temperatury otoczenia rzędu 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 pomiary co 350 ms, czyli około 3 pomiarów na sekundę. Zauważmy, że mierząc cztery niezależne sygnały (z czterech głośników) kolejny sygnał mierzony jest co 87 ms, dźwięk poruszający się z prędkością 346 m/s w tym czasie przebędzie odległość 29.7 metrów. Tak duży dystans w stosunku do powierzchni roboczej jest konieczny by zapobiec nakładaniu się sygnałów pochodzących z poprzednich pomiarów z nowymi (nadany sygnał odbija się od ścian i wpływa na kolejne pomiary, czas rzędu 87 ms pozwala na jego całkowite rozproszenie). Można temu zaradzić nadając za każdym razem inny rozróżnialny sygnał, to podejście wymaga jednak nadajników i odbiorników o większych możliwościach wysterowania.

- Stosunkowo mały kąt widzenia. Zastosowane przetworniki piezoelektryczne są przetwornikami kierunkowymi, co znacząco wpływa na maksymalny kąt pod jakim można ustawić nadajnik względem odbiornika. W prezentowanym prototypie kąt ten nie powinien przekraczać 45°. Zastosowanie większej liczby przetworników lub inny ich rodzaj powinien znacząco zwiększyć ów kąt.

- Ograniczone możliwości wysterowania rezonatorów piezoelektrycznych. Zastosowane rezonatory nie nadają się do generowania krótkich impulsów, co znacząco uprościłyby mierzenie odległości.

- Problem z pomiarem pozycji i orientacji nadajnika w ruchu. W prezentowanym urządzeniu jeden pomiar pozycji i orientacji wymaga pomiaru położenia czterech głośników w różnych odstępach czasowym. Jeśli podczas takiego pomiaru nadajnik porusza się z większą prędkością to otrzymane położenia przesunięte są względem siebie zgodnie z kierunkiem ruchu nadajnika. Takie przesunięcie negatywnie wpływa na pomiar orientacji.

Jak widać, wykorzystanie dźwięków to określania pozycji i orientacji prócz niewątpliwych zalet ma również wiele wad. Najistotniejszą wadą wydaje się być duża wrażliwość na warunki atmosferyczne a szczególnie zmianę temperatury. Jest to zapewne główny powód dlaczego ów rozwiązanie nie jest szeroko stosowane w produktach komercyjnych.

Bibliografia

[1] Dean, E. A., *Atmospheric Effects on the Speed of Sound*, BATTELLE COLUMBUS LABS DURHAM NC, AUG 1979
http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA076060

[2] Marcin Peczarski, *USB dla niewtajemniczonych w przykładach na mikrokontrolery STM32*, Wydawnictwo BTC, Legionowo 2013,
http://www.mimuw.edu.pl/~marpe/book/stm32_usb_101.zip

[3] Weisstein, Eric W. *Cross-Correlation Theorem.* From MathWorld–A Wolfram Web Resource.
http://mathworld.wolfram.com/Cross-CorrelationTheorem.html

[4] *Arduino Nano (v2.3) User Manual*
https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf

[5] *ULTRASONIC SENSOR (GENERAL), 40ST-12*
https://www.maritex.com.pl/media/uploads/PRODUKTY_PDF/sens/40str-12.pdf

[6] Discovery kit for STM32F407/417 lines,
http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/PF252419

[7] datasheet: STM32F405xx, STM32F407xx
http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/DM00037051.pdf

[8] James Karki, *Signal Conditioning Piezoelectric Sensors*, Application Report: SLOA033A - September 2000
http://www.ti.com/lit/an/sloa033a/sloa033a.pdf

[9] *Dual low-noise operational amplifiers.* SLOS075I, november 1979, revised april 2009
http://www.ti.com/lit/ds/symlink/ne5532.pdf

[10] *TLV277x, TLV277xA - Family of 2.7-V high-slew-rate rail-to-rail output operation amplifiers with shutdown.* SLOS209G - january 1998 - revised february 2004
http://www.ti.com/lit/ds/symlink/tlv2774.pdf

[11] Texas Instruments, *LM341/LM78MXX Series 3-Terminal Positive Voltage Regulators* SNVS090E - MAY 2004 - REVISED AUGUST 2005
http://www.ti.com/lit/ds/symlink/lm341.pdf

[12] datasheet, *ATmega48A/PA/88A/PA/168A/PA/328/P - ATMEL 8-BIT MICROCONTROLLER WITH 4/8/16/32KBYTES IN-SYSTEM PROGRAMMABLE FLASH*

Atmel 2014
http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88P
datasheet_Complete.pdf

[13] *Arduino Software* Arduino
https://www.arduino.cc/en/Main/Software

[14] Parth Rajesh Desai, Pooja Nikhil Desai, Komal Deepak Ajmera, Khushbu Mehta, *A Review Paper on Oculus Rift-A Virtual Reality Headset*
http://arxiv.org/pdf/1408.1173.pdf

[15] vrwiki, *Oculus Rift Development Kit 2*
https://vrwiki.wikispaces.com/Oculus+Rift+Development+Kit+2

[16] Technical Illusions, *Press Kit*
http://technicalillusions.com/wp-content/uploads/2013/10/Press-Kit.pdf

[17] Wenjun Zeng, *Microsoft Kinect Sensor and Its Effect*, Multimedia at Work
http://research.microsoft.com/en-us/um/people/zhang/Papers/Microsoft%
20Kinect%20Sensor%20and%20Its%20Effect%20-%20IEEE%20MM%202012.pdf

[18] Changchang Wu, *VisualSFM : A Visual Structure from Motion System*
http://ccwu.me/vsfm/

[19] Christian Banker, Michael Cretella, Jeff Dimaria, Jamie Mitchell, Jeff Tucker, *Ultrasonic 3D Wireless Computer Mouse*, Worcester Polytechnic Institute Electrical and Computer Engineering Department
https://www.wpi.edu/Pubs/E-project/Available/E-project-042607-125024/
unrestricted/magicmouse_2007-04-26.pdf

[20] *The Pi 3D scanner project*
http://www.pi3dscan.com/

[21] *Power Glove*, wikipedia
https://en.wikipedia.org/wiki/Power_Glove

[22] *Reach Out And Touch Your Data*, BYTE JULY 1990, IBM
http://research.microsoft.com/en-us/um/people/bibuxton/buxtoncollection/
a/pdf/BYTE%20Glove%20Comparision.pdf

[23] *Kąty Eulera*. wikipedia
https://pl.wikipedia.org/wiki/K%C4%85ty_Eulera