

Kierunek: **Informatyka algorytmiczna (INA)**

PRACA DYPLOMOWA
MAGISTERSKA

**Mechanizm multilateracji w
rozproszonej sieci sensorów audio**

**Multilateration mechanism in
distributed net of audio sensors**

Gabriel Budziński

Opiekun pracy
dr inż. Przemysław Błaśkiewicz

Słowa kluczowe: multilateracja, sensory audio, synchronizacja czasu

Streszczenie

Problem pozycjonowania w przestrzeni na podstawie emitowanego dźwięku obiektu pozycjonowanego wiąże się z wykorzystaniem możliwie zsynchronizowanych w czasie węzłów (mikrofonów) i pomiarze różnic czasu odbioru dźwięku przez czujniki. W pracy zostanie zbudowana sieć (co najmniej 4 sztuki) sensorów audio połączonych bezprzewodowo między sobą i ze stacją główną. Zadaniem sieci będzie wskazanie lokalizacji w przestrzeni punkowego przedmiotu emitującego dźwięk. Oprócz wyboru i implementacji algorytmu multilateracji zaproponowane zostanie rozwiązanie problemu synchronizacji czasu między sensorami, minimalizacji opóźnień w komunikacji oraz kalibracji systemu.

Słowa kluczowe: multilateracja, sensory audio, synchronizacja czasu

Abstract

The problem of positioning in space based on the emitted sound of the positioned object involves the use of as closely synchronized nodes (microphones) as possible in time and measuring the differences in the time of sound reception by sensors. In the work, a network (of at least 4 units) of audio sensors connected wirelessly to each other and to the main station will be built. The network's task will be to indicate the location in space of a point-like object emitting sound. In addition to selecting and implementing the multilateration algorithm, a solution to the problem of time synchronization between sensors, minimizing communication delay, and system calibration will be proposed.

Keywords: multilateration, WASN, clock synchronization

Spis treści

1. Przedstawienie problemu	5
1.1. Aktualny stan wiedzy	6
2. Sprzęt systemowy	7
2.1. Serwer MQTT	7
2.2. Węzeł	8
2.3. Serwer obliczeniowy	8
3. Eksperyment zerowy	9
3.1. Opis działania	9
3.1.1. Program węzła	9
3.1.2. Program serwera	11
3.1.3. Opis algorytmu	11
3.2. Ewaluacja działania systemu	11
3.2.1. Struktura systemu testowego	11
3.3. Interpretacja wyników i wnioski	12
4. Synchronizacja czasu	14
4.1. Synchronizacja programowa	14
4.1.1. Algorytm synchronizacji NTP	14
4.1.2. Bezpośredni pomiar przesunięć zegarów	16
4.2. Synchronizacja sprzętowa	17
4.2.1. Synchronizacja z użyciem mikrofonów	17
4.2.2. Porównanie metod	18
5. Multilateriacja	20
5.1. Przygotowanie danych wejściowych	20
5.1.1. Korekcja odległości	20
5.2. Ewaluacja działania systemu	22
5.3. Wyniki	22
5.3.1. Interpretacja	22
5.3.2. Wnioski	22
Literatura	24

Wstęp

Multilateracja jest metodą lokalizacji opartą na odległościach od punktów o znanych współrzędnych do szukanego punktu aby określić jego położenie. Zazwyczaj odległości te uzyskuje się na podstawie czasu, w którym sygnał dotarł do odbiornika (*ang. time of arrival, TOA*). Problem tego typu jest podstawowym w wielu współczesnych serwisach lokalizacyjnych, takich jak GPS (*ang. Global Positioning System*) i był od badany przynajmniej od lat '60, ze znacznymi postępami poczynionymi w latach '70 podczas opracowywania systemu GPS, a w latach '90 w związku z błyskawicznym wzrostem liczby telefonów komórkowych i potrzebą lokalizacji osób komunikujących się przy ich użyciu z służbami ratunkowymi [1]. W ostatnich latach coraz większym zainteresowaniem cieszy się zastosowanie metod multilateracji w sieciach urządzeń IoT [2] w celu autonomicznego uzyskiwania informacji lokalizacji przestrzennej bez potrzeby użycia dedykowanej do tego celu aparatury.

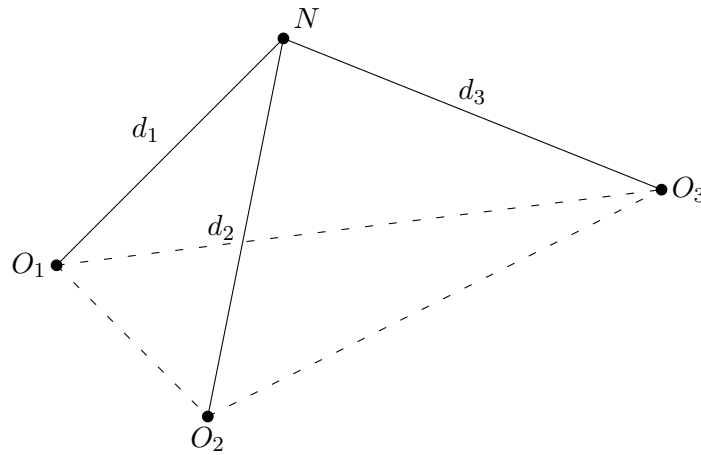
Na przestrzeni lat rozwijano wiele wariantów rozwiązań problemu lokalizacji na podstawie czasu otrzymania sygnału. Pierwszym były metody iteracyjne, dzięki którym w prosty sposób otrzymywano rozwiązanie równań nieliniowych, ale nie gwarantowały ani zbiegania do poprawnego rozwiązania, ani krótkiego czasu otrzymania go. W latach '80 wprowadzono metody manipulacji układami równań tak by otrzymać równanie kwadratowe jednej niewiadomej dające dwa rozwiązania, z których jedno było poprawne. Wreszcie w połowie lat '90 zaczęto używać układów równań liniowych, których rozwiązanie otrzymywano dzięki programowaniu liniowemu. Zaletą tej metody jest jej wszechstronność, ponieważ daje możliwość łatwego rozszerzenia problemu o więcej pomiarów. Te dodatkowe informacje mogą być dodatkowymi pomiarami TOA lub każdymi innymi związanymi z problemem przy pomocy równań liniowych. Dzięki temu mogły rozwijać się algorytmy lokalizacji wykorzystujące kąt przybycia (*ang. angle of arrival, AOA*), różnicę częstotliwości (*ang. frequency difference of arrival, FDOA*), moc odebranego sygnału (*ang. received signal strength, RSS*) oraz kierunek przybycia¹ (*ang. direction of arrival, DOA*). W poniższej pracy skupimy się na podstawowym wariancie multilateracji, czyli TOA.

¹Różnicą między DOA a AOA jest odległość, z której odbierany jest sygnał. DOA zakłada, że źródło sygnału jest na tyle daleko, że AOA jest wspólny dla wszystkich odbiorników. W AOA różnice pomiarów są na tyle duże, aby móc użyć ich do triangulacji.

Rozdział 1

Przedstawienie problemu

Multilateracja jest techniką lokalizacji pozwalającą obliczyć nieznane koordynaty punktu na podstawie odległości od innych, znanych punktów. Weźmy trójwymiarowy egzemplarz naszego problemu (Rys. 1.1), gdzie N - nadajnik, O_i - odbiorniki, d_i - odległości. Przedstawienie problemu zostało zaczerpnięte z pracy [3].



Rys. 1.1: Egzemplarz problemu multilateracji

Znalezienie koordynatów (x, y) punktu N jest równoważne z rozwiązaniem układu równań,

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 = d_1^2 \\ (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 = d_2^2 \\ (x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 = d_3^2 \end{cases} \quad (1.1)$$

który może zostać przekształcony do postaci

$$\begin{cases} (x^2 + y^2 + z^2) - 2x_1x - 2y_1y - 2z_1z = d_1^2 - x_1^2 - y_1^2 - z_1^2 \\ (x^2 + y^2 + z^2) - 2x_2x - 2y_2y - 2z_2z = d_2^2 - x_2^2 - y_2^2 - z_2^2 \\ (x^2 + y^2 + z^2) - 2x_3x - 2y_3y - 2z_3z = d_3^2 - x_3^2 - y_3^2 - z_3^2 \end{cases} \quad (1.2)$$

lub w reprezentacji macierzowej,

$$\begin{bmatrix} 1 & -2x_1 & -2y_1 & -2z_1 \\ 1 & -2x_2 & -2y_2 & -2z_2 \\ 1 & -2x_3 & -2y_3 & -2z_3 \end{bmatrix} \begin{bmatrix} x^2 + y^2 + z^2 \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} d_1^2 - x_1^2 - y_1^2 - z_1^2 \\ d_2^2 - x_2^2 - y_2^2 - z_2^2 \\ d_3^2 - x_3^2 - y_3^2 - z_3^2 \end{bmatrix} \quad (1.3)$$

którą można przedstawić jako

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b} \quad (1.4)$$

przy założeniu, że $\mathbf{x} \in E$, gdzie $E = \{(x_0, x_1, x_2, x_3)^T \in \mathbb{R}^4 \mid x_0 = x_1^2 + x_2^2 + x_3^2\}$.

Uogólniona forma równania macierzowego problemu multilateracji dla przestrzeni n -wymiarowej i m odbiorników:

$$\begin{bmatrix} 1 & -2\mathbf{x}^{(1)} \\ 1 & -2\mathbf{x}^{(2)} \\ \vdots & \vdots \\ 1 & -2\mathbf{x}^{(m)} \end{bmatrix} \begin{bmatrix} \sum_{i=1}^n x_i^2 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1^2 - \sum_{i=1}^n x_i^{(1)2} \\ d_2^2 - \sum_{i=1}^n x_i^{(2)2} \\ \vdots \\ d_m^2 - \sum_{i=1}^n x_i^{(m)2} \end{bmatrix} \quad (1.5)$$

1.1. Aktualny stan wiedzy

W dostępnej literaturze obejmującej temat multilateracji i systemów multilateracyjnych opisanych jest wiele podejść do tego problemu:

- układ równań liniowych[4],
- metoda najmniejszej sumy kwadratów[4],[3],
- nieliniowa metoda najmniejszej sumy kwadratów[4],
- rozkład według wartości szczególnych[4].

Na szczególną uwagę zasługuje praca [4], w której przeprowadzono eksperymenty porównawcze wyżej wymienionych metod.

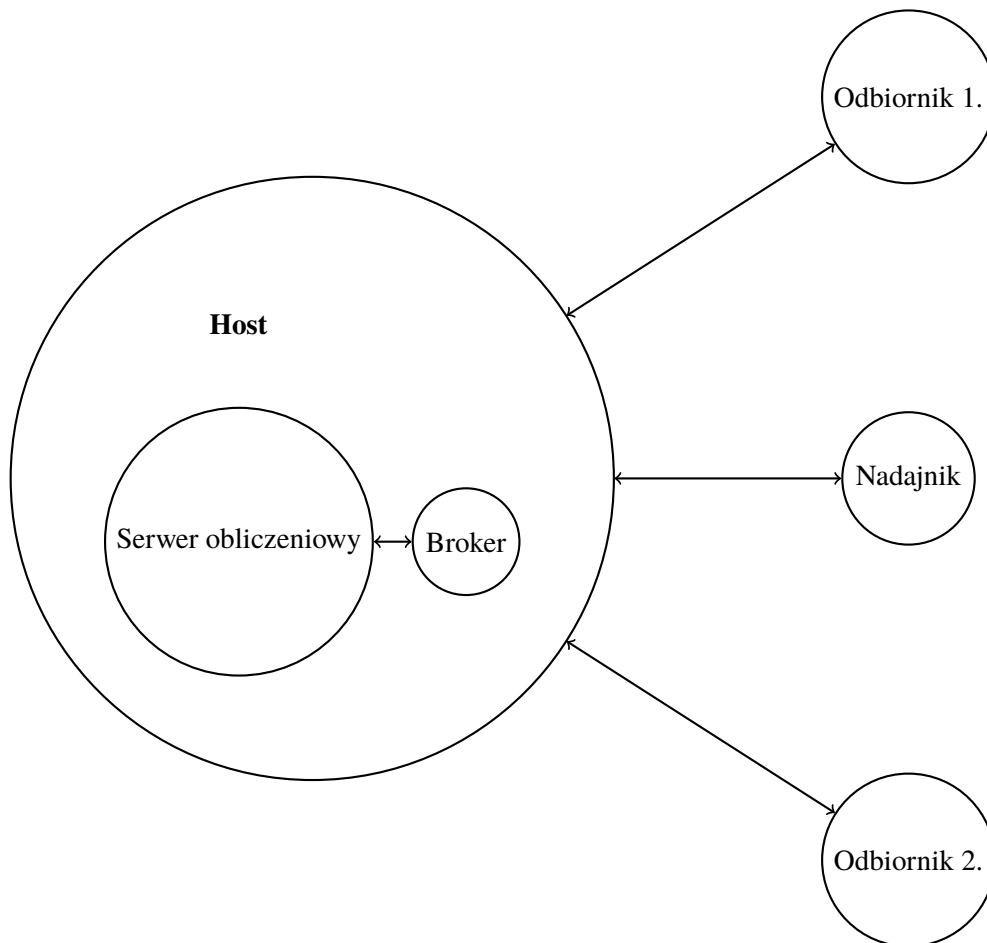
Motywacją tej pracy jest brak dostępnych prac obejmujących to zagadnienie w spektrum fal dźwiękowych, w przeciwieństwie do fal elektromagnetycznych, an których opierają się systemy opisane w pozycjach bibliograficznych. Pytania, które zostały postawione:

-

Rozdział 2

Sprzęt systemowy

Znaczna większość prac adresujących temat multilateracji opiera się na systemach urządzeń działających w zakresie fal elektromagnetycznych ([4],[3],[5]). W tej pracy poświęcimy uwagę systemowi działającemu w domenie dźwięku, jak ten aspekt wpływa na skuteczność i dokładność rozwiązania problemu multilateracji.



Rys. 2.1: Topologia systemu

2.1. Serwer MQTT

Urządzenia systemowe porozumiewają się przy użyciu protokołu MQTT, każdy z węzłów oraz serwer łączą się z centralnym brokerem (?), który przekierowuje wiadomości do klientów, które zasubskrybowały dany temat.

2.2. Węzeł

Każdy z węzłów oparty jest o mikrokontroler ESP8266 zaprogramowany przy użyciu Arduino IDE. W systemie występują dwa rodzaje węzłów:

- nadajnik,
- odbiornik.

Nadajnik jest wyposażony jest w przełącznik cewkowy sterowany przez mikrokontroler, który służy do kontrolowania brzęczyka zasilanego napięciem $12V$. Wybrano brzęczyk o głośności $90dB$ w celu zmaksymalizowania zasięgu działania systemu.

Odbiornik natomiast wyposażony jest w mikrofon elektretowy dającego binarny sygnał wyjściowy. Czułość mikrofonu dostrajana jest ręcznie poprzez potencjometr.

2.3. Serwer obliczeniowy

Centralnym urządzeniem systemu jest serwer obliczeniowy kumulujący dane otrzymane z sensorów do rozwiązania problemu multilateracji. Serwer hostowany jest na tej samej maszynie co broker MQTT.

Rozdział 3

Eksperyment zerowy

Po przygotowaniu komponentów systemu wstępnie zaimplementowano program rozwiązujący problem multilateracji, aby zbadać, czy problem nie jest zbyt trywialny, aby opisać go w pracy, lub przeciwnym razie, na podstawie wyników eksperymentu zastanowić się jakie przeszkody stoją na drodze do rozwiązania o zadowalającej precyzji.

3.1. Opis działania

Program zaimplementowano na podstawie rozwiązania aproksymacyjnego równania 1.4 postaci

$$\hat{\mathbf{x}} = \left(A^T A \right)^{-1} A^T \mathbf{b} \quad (3.1)$$

za pomocą metody najmniejszych kwadratów zaczerpniętej z artykułu [3]. Współrzędne odbiorników są znane.

3.1.1. Program węzła

Algorithm 1 Program nadajnika

```
1: buzz  $\leftarrow$  False
2: buzzTime  $\leftarrow$  0
3: lastBuzzTime  $\leftarrow$  0
4: syncTime  $\leftarrow$  0
5: function ONMESSAGE(message, topic)
6:   if topic is TOPIC then
7:     buzz  $\leftarrow$  message
8:   end if
9:   if topic is TIME_TOPIC then
10:    syncTime  $\leftarrow$  micros()
11:   end if
12: end function
13: loop
14:   if buzz and micros() – lastBuzzTime > BUZZ_INTERVAL then
15:     buzzTime  $\leftarrow$  micros() – syncTime
16:     publish(buzzTime)
17:     lastBuzzTime  $\leftarrow$  buzzTime
18:     buzzer()
19:   end if
20: end loop
```

- *micros*() - funkcja zwracająca liczbę mikrosekund od uruchomienia urządzenia,

Algorithm 2 Program odbiornika

```
1:  $micState \leftarrow LOW$ 
2:  $micTime \leftarrow False$ 
3:  $lastMicTime \leftarrow 0$ 
4:  $syncTime \leftarrow 0$ 
5: function ONMESSAGE( $topic$ )
6:   if  $topic$  is  $TIME\_TOPIC$  then
7:      $syncTime \leftarrow micros()$ 
8:   end if
9: end function
10: function MICINTERRUPT
11:    $micState \leftarrow HIGH$ 
12: end function
13: loop
14:   if  $micState$  is  $HIGH$  and  $micros() - lastMicTime > MIC\_INTERVAL$  then
15:      $micTime \leftarrow micros() - syncTime$ 
16:      $publish(micTime)$ 
17:      $lastMicTime \leftarrow micTime$ 
18:      $micState \leftarrow LOW$ 
19:   end if
20: end loop
```

- *TOPIC* - nazwa kanału MQTT korespondującego do danego węzła,
- *TIME_TOPIC* - nazwa kanału MQTT służącego do przesyłania wiadomości o synchronizacji czasu.
- *publish()* - funkcja publikująca wiadomość na zadanym kanale MQTT

3.1.2. Program serwera

Algorithm 3 Program serwera

```

1:  $M \leftarrow (A^T A)^{-1} A^T$ 
2: function calc_position
3:   for node in NODES do
4:      $d \leftarrow (SS/10^6) \cdot \text{time}(\text{node}) - \text{time}(\text{source})$ 
5:      $b_{\text{node}} \leftarrow d - \sum \text{coords}(\text{node})^2$ 
6:   end for
7:   return  $M \cdot b$ 
8: end function
9: loop main
10:   sleep( $t_1$ )
11:   calc_position()
12: end loop
13: loop sync_clock
14:   publish(TIME_TOPIC)
15:   sleep( $t_2$ )
16: end loop

```

- A - macierz zawierająca współrzędne obrotów z równania 1.4
- SS - prędkość dźwięku $[\frac{m}{s}]$,
- *time()* - czas dostarczony w ostatniej wiadomości od węzła,
- *coords()* - współrzędne węzła,
- t_1 - czas pomiędzy obliczeniami pozycji,
- t_2 - czas pomiędzy wiadomościami synchronizującymi zegary.

3.1.3. Opis algorytmu

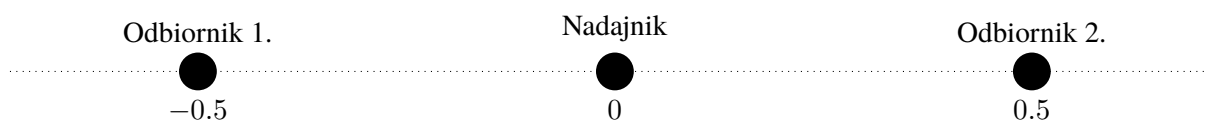
W programie serwera zaimplementowano funkcję rozwiązującą równość 3.1 przy pomocy funkcji bibliotecznych dostępnych w pakiecie *numpy*. Jak łatwo zauważyć pierwsza część lewej strony równości zależy jedynie od macierzy A , która nie zawiera zmiennych związanych z węzłem źródłowym, co pozwala na wcześniejsze i jednorazowe przeprowadzenie kosztownych przekształceń tej macierzy. W tej sytuacji jednorazowe wywołanie funkcji wymaga jedynie obliczenia wektora b oraz iloczynu skalarnego. Wywołania następują co interwał t_1 . Co interwał t_2 serwer wysyła wiadomość synchronizującą z założeniem, że węzły odbierają ją równocześnie i w ten sposób synchronizują zegary.

3.2. Ewaluacja działania systemu

Pierwsze testy przeprowadzone zostały na systemie operującym w jednym wymiarze (to jest wszystkie węzły są współliniowe) w celu jak największego uproszczenia, pozwalającego na szybsze wykrywanie i rektyfikację błędów. Wszystkie odległości będą podawane w metrach.

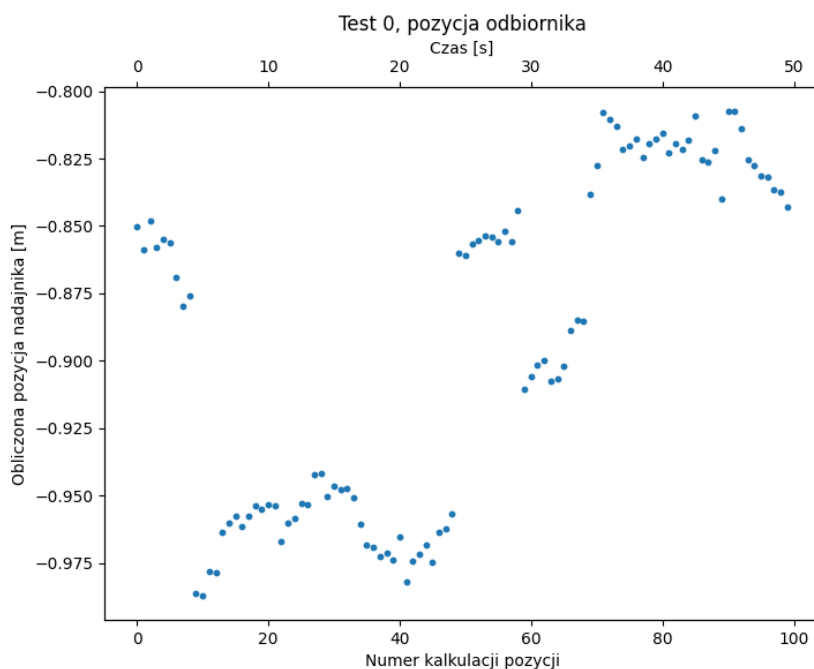
3.2.1. Struktura systemu testowego

Nadajnik umiejscowiono w punkcie (0), natomiast dwa odbiorniki w punktach odpowiednio (−0.5) i (0.5), wszystkie węzły były stacjonarne. Nadajnik co 0.5 s nadawał sygnał o długości 10 ms, a serwer co



Rys. 3.1: Układ systemu testowego

0.5 s zwracał wynik zagadnienia multilateracji na podstawie ostatnio otrzymanych danych. Wykonano 5 następujących bezpośrednio po sobie eksperymentów, poniżej przedstawiono wyniki trzech pierwszych, ponieważ są wystarczające do ukazania zachodzącego trendu.



Rys. 3.2: Wykres obliczonej pozycji odbiornika w zależności od czasu

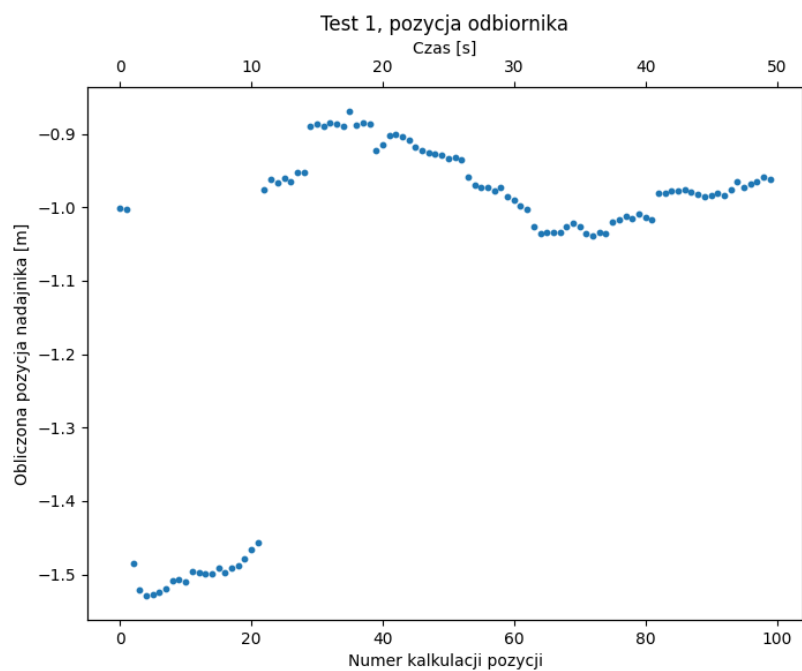
3.3. Interpretacja wyników i wnioski

Już na pierwszym z wykresów widać, że wartości wyjściowe algorytmu są niestabilne, współrzędna nadajnika waha się w przedziale $(-0.98, -0.8)$, a także co jakiś czas następują nagłe przeskoki między następującymi po sobie wartościami. Coraz większa rozbieżność następuje wraz z upływem czasu. Już w drugim teście różnica ekstremalnych odchyleń jest większa niż odległość między nadajnikami, a dokładność jest absolutnie niezadowalająca. Kolejne testy jedynie utwierdzają te obserwacje.

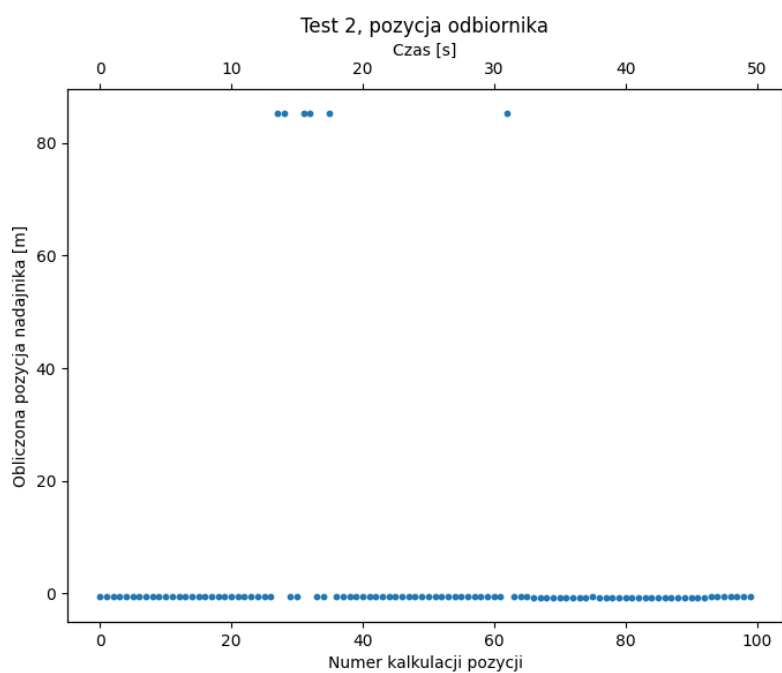
Na niską dokładność i stabilność wyników może mieć wpływ wiele czynników takich jak:

- niepoprawna synchronizacja zegarów węzłów,
- niepoprawna kalibracja czułości mikrofonów,
- fałszywe odczyty wynikające z odbić fali dźwiękowej,
- niska jakość rozwiązania aproksymacyjnego.
- TODO

Obserwując skoki pomiędzy sąsiadującymi obserwacjami położenia rozdzielającymi okresy względnej stabilności i mając na uwadze statyczność otoczenia systemu możemy śmiało wysnuć wniosek, że tak nagła zmiana średniej obliczanej wartości może wynikać najprawdopodobniej z błędnej synchronizacji czasu. Wygląda na to, że sygnał wysyłany przez serwer obliczeniowy nie jest odbierany we wszystkich



Rys. 3.3: Wykres obliczonej pozycji odbiornika w zależności od czasu



Rys. 3.4: Wykres obliczonej pozycji odbiornika w zależności od czasu

węzłach równocześnie. Następny rozdział będzie poświęcony badaniu i tworzeniu algorytmu synchronizacji zegarów w węzłach.

Rozdział 4

Synchronizacja czasu

Synchronizacja czasu jest jednym z podstawowych problemów systemów opartych o wiele urządzeń posiadających własne zegary. Wiedza o dokładnym czasie zachodzących w sieci zdarzeń jest niezbędna do szybkiego przesyłu danych, koordynacji procesów czy aktualizacji systemu plików. W przypadku niniejszej pracy dokładne określenie czasu zachodzących zdarzeń jest kluczowe do wiarygodnego określenia odległości pomiędzy węzłami na podstawie interwału czasowego pomiędzy nadaniem a odbiorem sygnału dźwiękowego.

4.1. Synchronizacja programowa

Pierwszym podejściem do rozwiązania problemu synchronizacji zegarów było zastosowanie synchronizacji programowej [6], w której urządzenie-host utrzymuje wysokiej rozdzielczości licznik czasowy i wysyła sygnały o jego wartości pozostałym węzłom w sieci. Na podstawie tych informacji każdy z węzłów oblicza różnicę w zegarach i wprowadza odpowiednie przesunięcie własnego licznika.

4.1.1. Algorytm synchronizacji NTP

Jednym z najbardziej rozpowszechnionych protokołów synchronizacji programowej jest *Network Time Protocol* opisany w pracy [7]. Zachowując oznaczenia w niej zastosowane opiszmy w skrócie zasadę działania tego protokołu.

Na rysunku 4.1 przedstawiono schemat działania protokołu NTP. Urządzenia A i B wymieniają wiadomości zawierające sygnatury czasowe. Niech T_i , T_{i-1} , T_{i-2} , T_{i-3} będą czterema ostatnimi wiadomościami oraz niech $a = T_{i-2} - T_{i-3}$ oraz $b = T_{i-1} - T_i$. Wtedy całkowity czas transmisji δ_i i przesunięcie zegara θ_i urządzenia B względem urządzenia A to

$$\delta_i = a - b \quad \text{oraz} \quad \theta_i = \frac{a + b}{2}$$

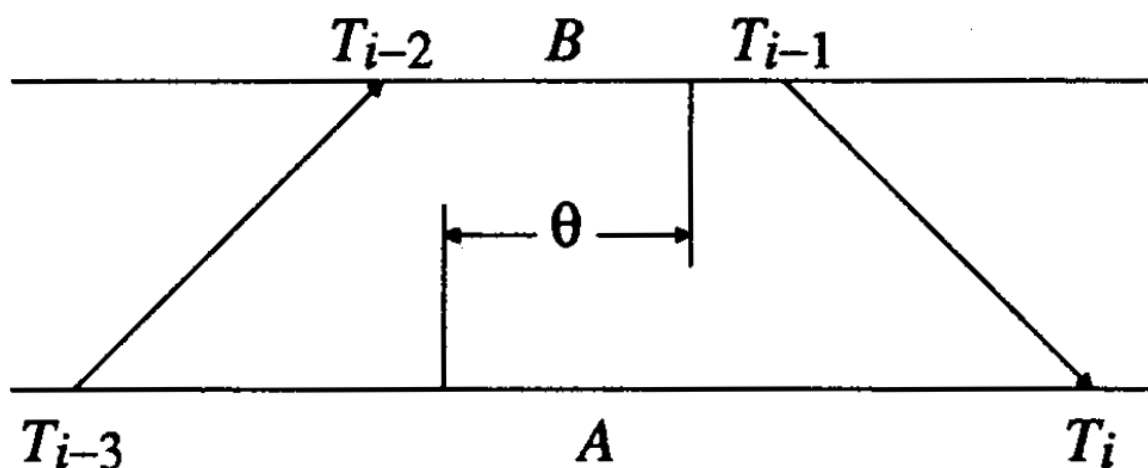
Dodatkowym wnioskiem przedstawionym w powyższej pracy jest własność prawdziwego przesunięcia względem aktualnie obliczonego:

$$\theta_i - \frac{\delta_i}{2} \leq \theta \leq \theta_i + \frac{\delta_i}{2}$$

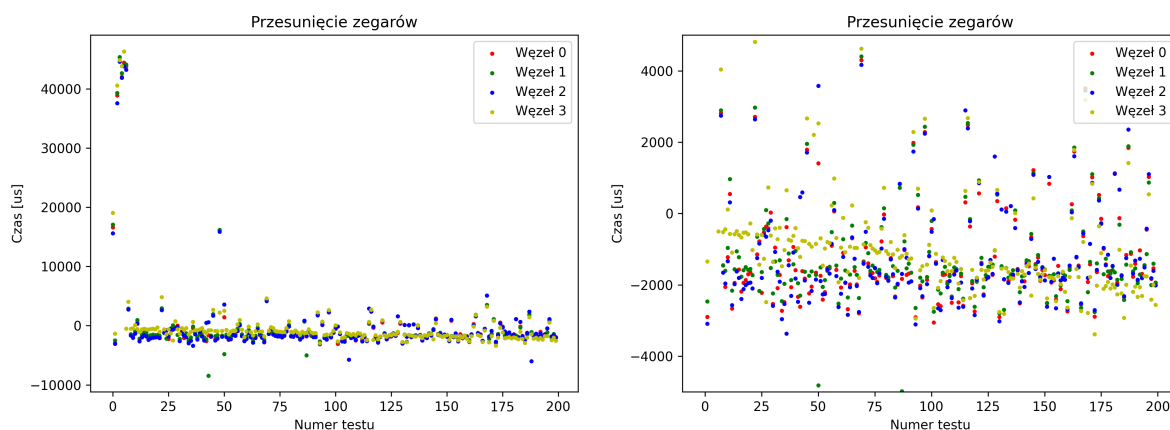
Jak łatwo zauważyć im krótszy jest czas propagacji tym lepsze przybliżenie dostajemy.

Na podstawie tych informacji podjęto próbę synchronizacji programowej zegarów węzłów w systemie multilateracyjnym. Przy użyciu czterech węzłów przeprowadzono eksperymenty mające na celu próbę ustalenia przesunięć zegarów każdego z urządzeń względem centralnego serwera. Opisany wyżej schemat wymiany czterech wiadomości powtarzano n razy wspólnie dla wszystkich węzłów zapisując obliczone przesunięcia oraz czasy propagacji.

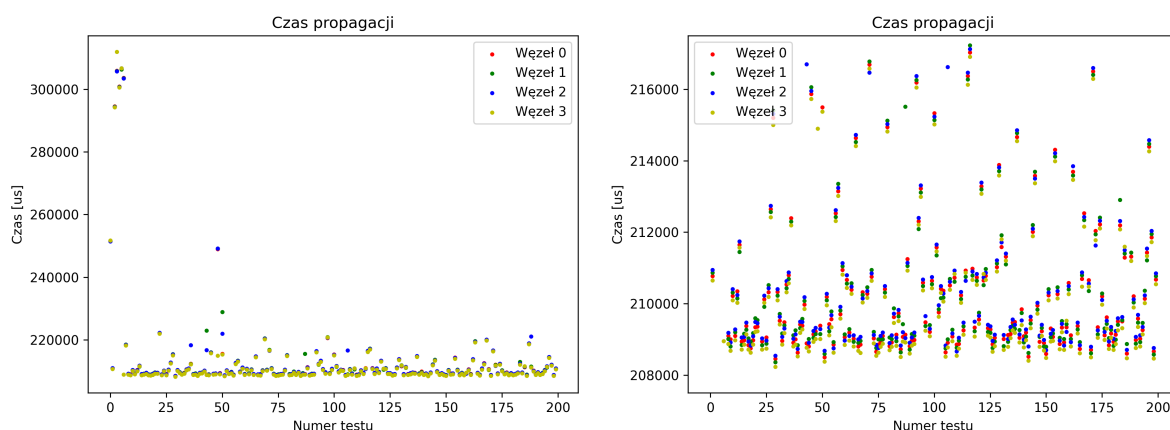
W celu poprawy czytelności wyniki przesunięć zostały znormalizowane poprzez przesunięcie o wartość średnią dla każdego z węzłów.



Rys. 4.1: Pomiar opóźnienia transmisji i przesunięcia zegara

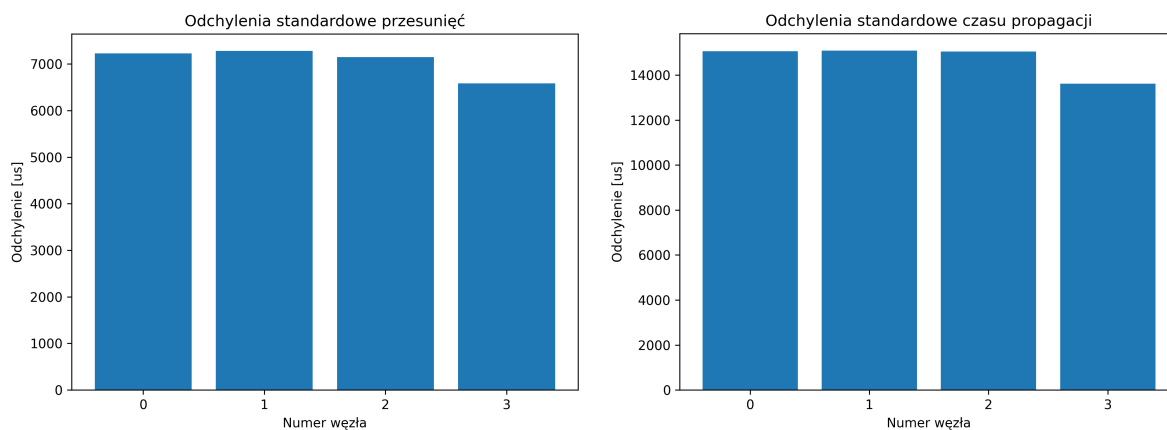


Rys. 4.2: Wyniki pomiarów przesunięć zegarów



Rys. 4.3: Wyniki pomiarów czasów propagacji

Na podstawie otrzymanych wykresów łatwo zauważyć, że czas propagacji wiadomości w systemie jest nieprzewidywalny i zmienia się znacząco z pomiaru na pomiar. Interesującą nas statystyką są jednak zaobserwowane przesunięcia zegarów, bez których nie jesteśmy w stanie poprawnie ocenić odległości od źródła dźwięku. Tutaj wartości wyglądają na bardziej skoncentrowane, jednak nie widać wyraźnych tendencji koncentracji wokół wartości średnich dla żadnego z węzłów. Ponadto zaobserwowane odchylenie

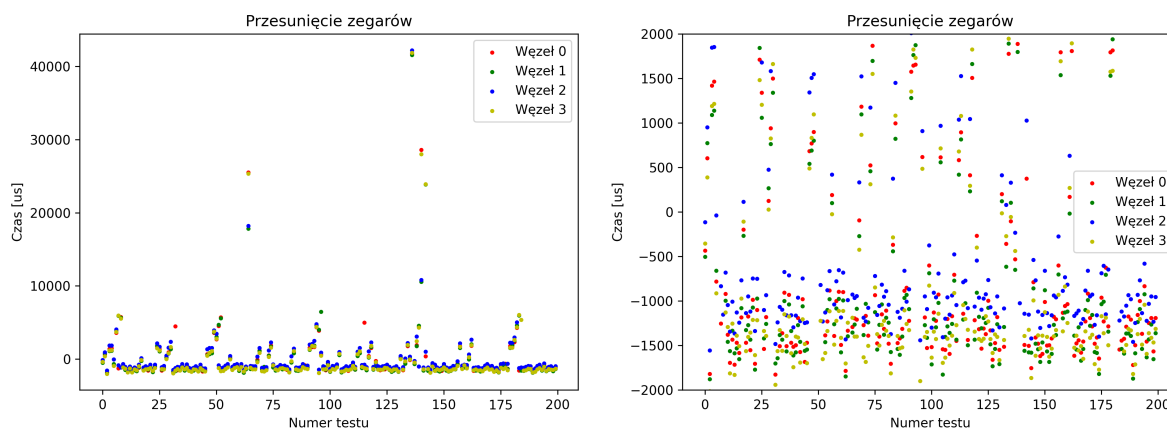


Rys. 4.4: Odchylenia standardowe pomiarów

nia standardowe σ_i , $i \in \{0, 1, 2, 3\}$ wielkości $\approx 7000\mu s$ są nieakceptowalne, ponieważ w takim czasie dźwięk w powietrzu pokonuje $\frac{7000}{1000000}s \cdot 343\frac{m}{s} = 2.401m$. Możliwe jest, że wielokrotne powtarzanie pomiarów da zadowalającą wartość średnią, pozwalającą na centymetrową precyzję obliczanych odległości. W porównaniu z pozostałymi sposobami brane będą pod uwagę wartości uśrednione.

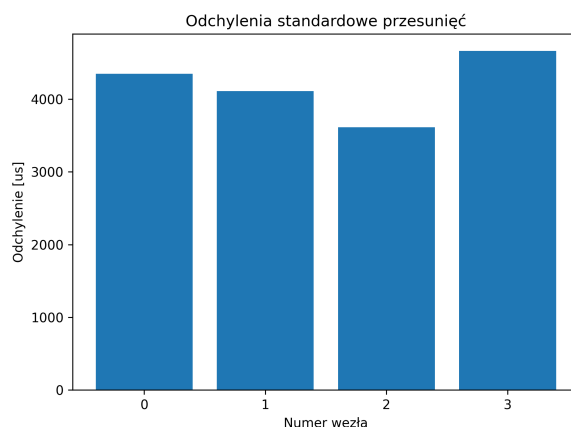
4.1.2. Bezpośredni pomiar przesunięć zegarów

Biorąc pod uwagę zauważoną nieprzewidywalność i rozrzut czasów propagacji (spowodowanych najprawdopodobniej użyciem protokołu MQTT do przesyłu wiadomości pomiędzy urządzeniami) następnym pomysłem schematu synchronizacji jest bezpośrednie badanie względnego przesunięcia zegarów. Węzeł wysyła n wiadomości zawierających aktualną wartość zegara, która po odebraniu przez serwer jest porównywana z zegarem w nim dostępnym.



Rys. 4.5: Wyniki pomiarów przesunięć zegarów

Wykresy przesunięć wygenerowane na podstawie tych testów na pierwszy rzut oka są skoncentrowane podobnie jak poprzednie, jednakże odchylenie standardowe są prawie dwukrotnie mniejsze niż uprzednio, co daje nadzieje na bardziej wiarygodne wyniki. W porównaniu z pozostałymi sposobami brane będą pod uwagę wartości uśrednione.



Rys. 4.6: Odchylenia standardowe pomiarów

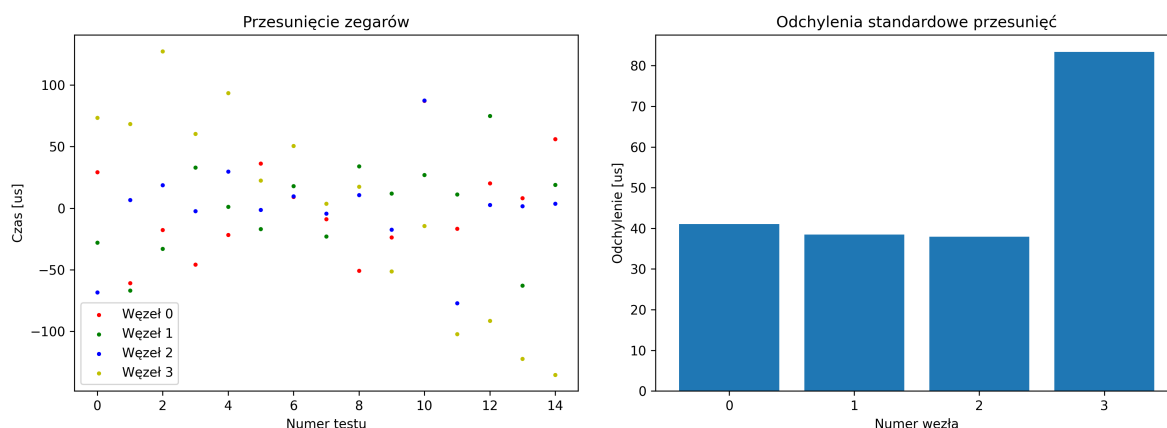
4.2. Synchronizacja sprzętowa

4.2.1. Synchronizacja z użyciem mikrofonów

Innym pomysłem na synchronizację zegarów w węzłach było użycie mikrofonów, w które węzły odbiorcze są wyposażone. Potrzebujemy znać jedynie przesunięcie naszego zegara względem zegara w węźle nadawczym dlatego wystarczającym będzie porównanie czasu nadania i odebrania sygnału dźwiękowego. Ponadto ten rodzaj synchronizacji w przeciwieństwie do synchronizacji programowej, która uzgadniała ze sobą jedynie zegary na podstawie wymienianych wiadomości, wlicza w czas transmisji wszelkie nie wzięte wcześniej pod uwagę opóźnienia, takie jak:

- Czas pomiędzy wysłaniem wiadomości o nadaniu sygnału a zamknięciem kontaktora i poruszeniem membraną brzęczyka,
- Czas pomiędzy odebraniem sygnału przez mikrofon a zmianą stanu zmiennej na to wskazującej.

Przeprowadzono testy tego typu synchronizacji, których wyniki przedstawiono na wykresach 4.7. W celu zwiększenia czytelności odrzucono pierwszy z pomiarów oraz przesunięto wyniki o wartość średnią.

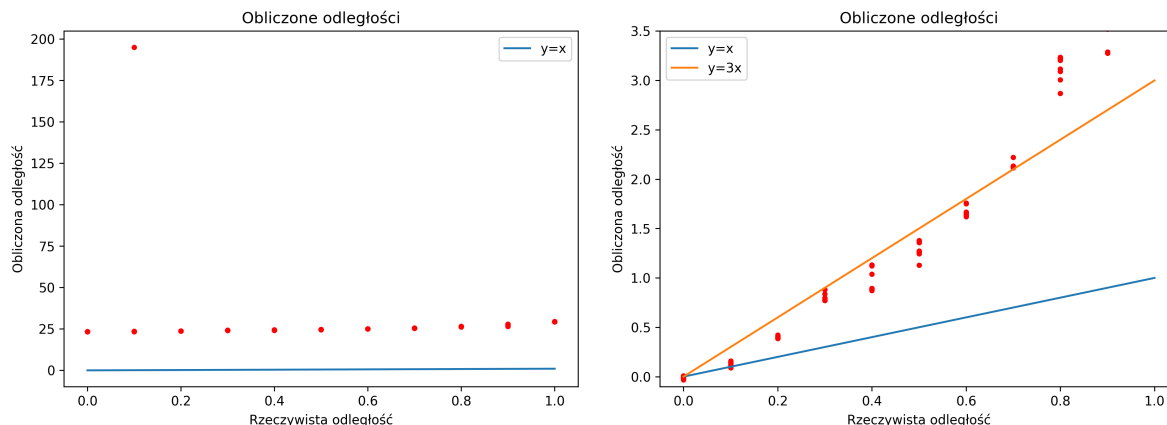


Rys. 4.7: Wyniki pomiarów przesunięć zegarów

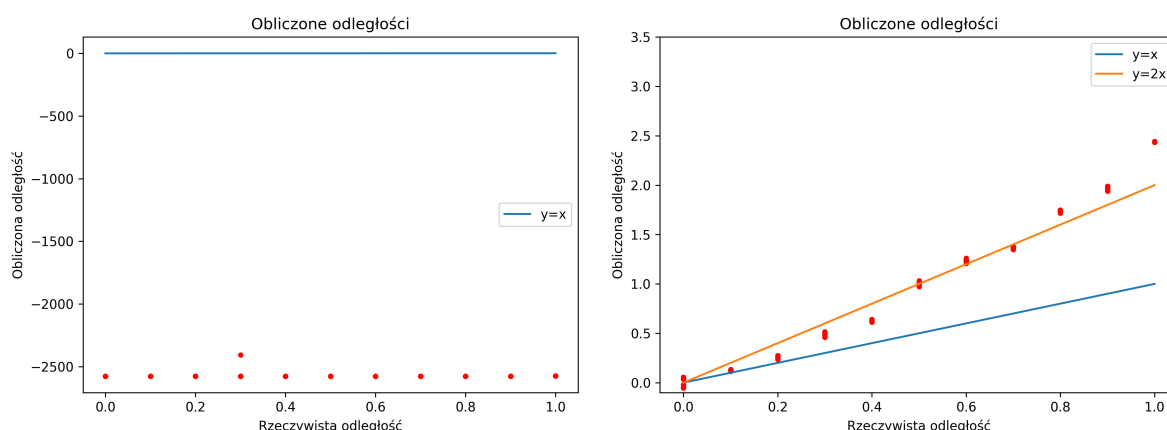
Łatwo zauważyć, że udało się zmniejszyć odchylenia obliczonych przesunięć zegara aż o dwa rzędy wielkości. Taka dokładność daje znacznie lepsze przybliżenie rzeczywistej odległości między węzłami, ponieważ w czasie $40\mu s$ dźwięk pokona jedynie $\frac{40}{10000000} s \cdot 343 \frac{m}{s} \approx 0,014m$. Mając tak dokładne odległości będziemy mogli wprowadzić je do modelu multilateracyjnego.

4.2.2. Porównanie metod

Porównajmy teraz dokładność obliczanych odległości na podstawie interwału czasowego pomiędzy nadaniem dźwięku a jego odbiorem w różnych wariantach synchronizacji zegarów. Wykresy uzyskane przy zastosowaniu synchronizacji czasowej zostały przedstawione przed i po znormalizowaniu poprzez przesunięcie tak, by średnia pomiarów rozpoczynała się od 0. Pomiary oparte o synchronizację z użyciem mikrofonów nie wymagały tego dodatkowego kroku.

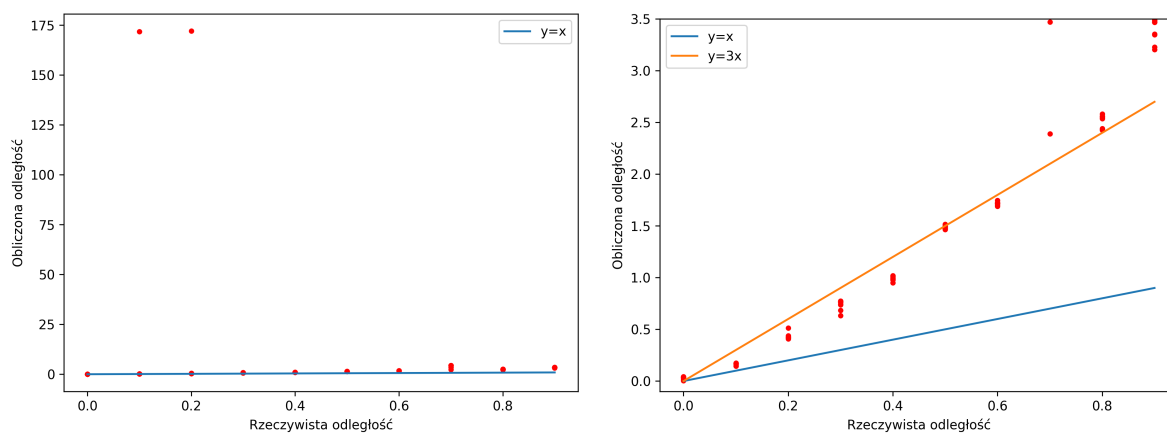


Rys. 4.8: Wyniki pomiarów odległości z użyciem synchronizacji 4.1.1



Rys. 4.9: Wyniki pomiarów odległości z użyciem synchronizacji 4.1.2

Na wykresach widać, że wszystkie trzy algorytmy synchronizacji po normalizacji dają wyniki o podobnej dokładności adekwatnej do użycia w danych wejściowych multilateracji. Ciekawą jest natomiast obserwacja skalowania obliczonych odległości. Widzimy, że w przypadku 4.8 oraz 4.10 otrzymane wartości leżą blisko trzykrotności rzeczywistej badanej odległości, natomiast w 4.9 blisko dwukrotności. W kolejnym rozdziale pochylimy się nad możliwymi przyczynami i rozwiązaniem tego zachowania.



Rys. 4.10: Wyniki pomiarów odległości z użyciem synchronizacji 4.2.1

Rozdział 5

Multilateriacja

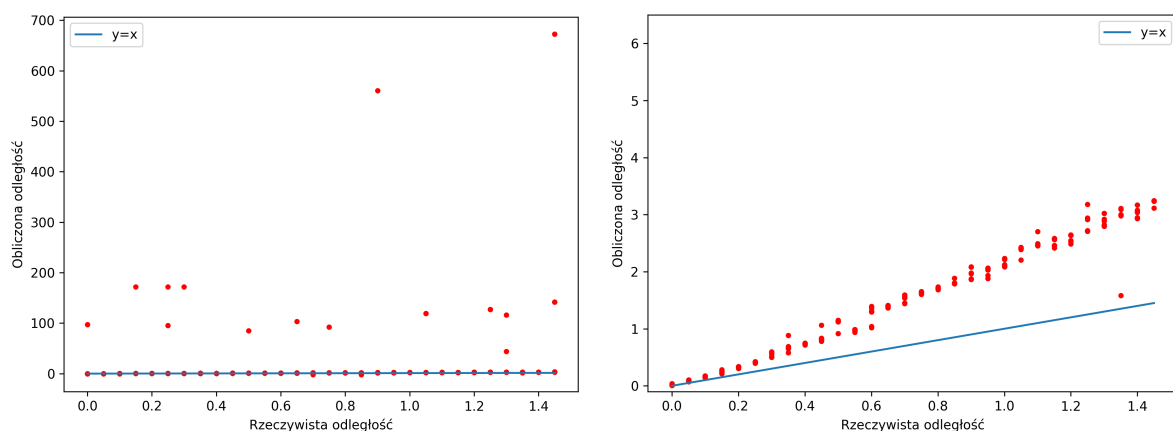
5.1. Przygotowanie danych wejściowych

Aby otrzymać poprawne wyniki algorytmu multilateracji dane wejściowe, w naszym przypadku odległości między węzłami odbiorczymi a nadajnikiem powinny być jak najbliższe rzeczywistym odległościom z możliwie małymi odchyleniami. Będziemy kontynuować usprawnianie metod uzyskiwania poprawnych wyników zapoczątkowane w rozdziale poprzednim.

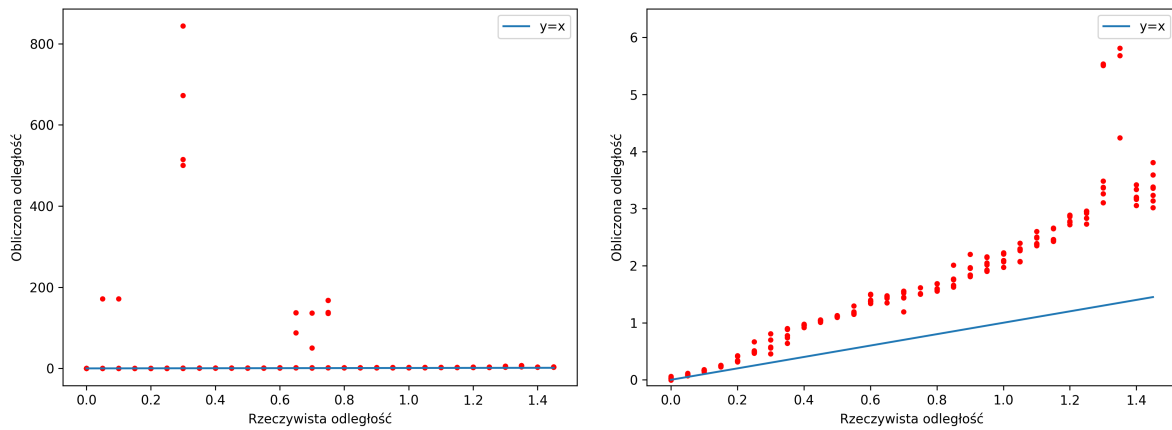
5.1.1. Korekcja odległości

W wynikach eksperymentów porównawczych metod synchronizacji czasu węzłów przeprowadzonych w rozdziale 4. zaobserwowaliśmy skalowanie wyników na pierwszy rzut oka zachowujące się liniowo. Przyjrzyjmy się teraz dokładnie temu zjawisku. W tym i kolejnych przypadkach będziemy używać już jedynie synchronizacji sprzętowej z użyciem mikrofonów ze względu na brak konieczności dodatkowej kalibracji przesunięcia punktu 0.

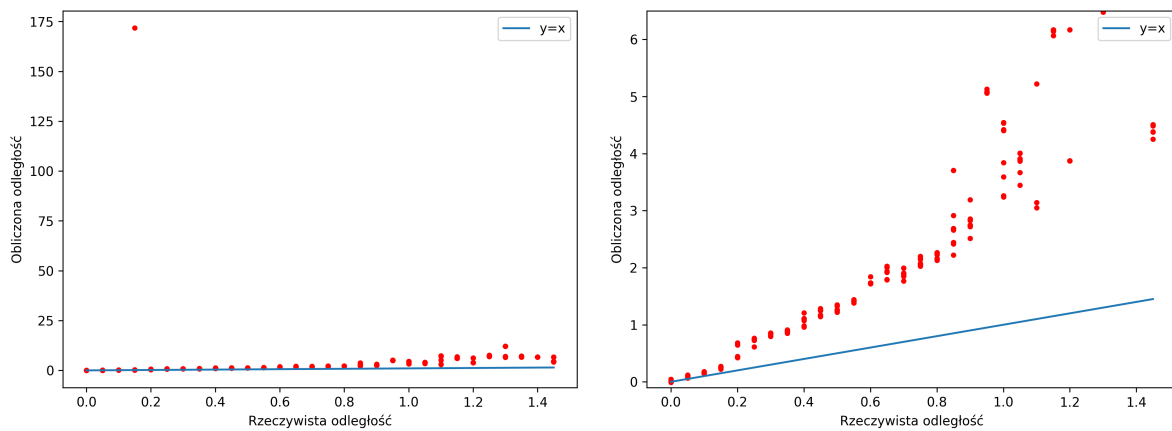
Prawdopodobnym powodem skalowania obliczanych odległości może być odczyt zmiany sygnału mikrofonowego odczytywanego przez mikrokontroler. Poniższe wykresy są wynikiem czterech kolejnych eksperymentów, w których jedyną zmienną była czułość zintegrowanego wzmacniacza mikrofonu. Wzmacniacz ten nie pozwala na precyzyjną regulację, a jedynie na zmianę rezystancji wbudowanego potencjometru. Ponieważ przedział czułości odpowiadający wykrywaniu sygnału węzła nadającego przy jednoczesnym zminimalizowaniu fałszywych aktywacji jest niewielki (około $\frac{1}{8}$ obrotu potencjometru) cztery zbadane przypadki nie dzielą równo badanego zakresu. Rozpoczynając od największej możliwej czułości przy każdym kolejnym eksperymencie zmniejszano ją póki pozwalała wciąż na wykrywanie sygnału z badanych odległości.



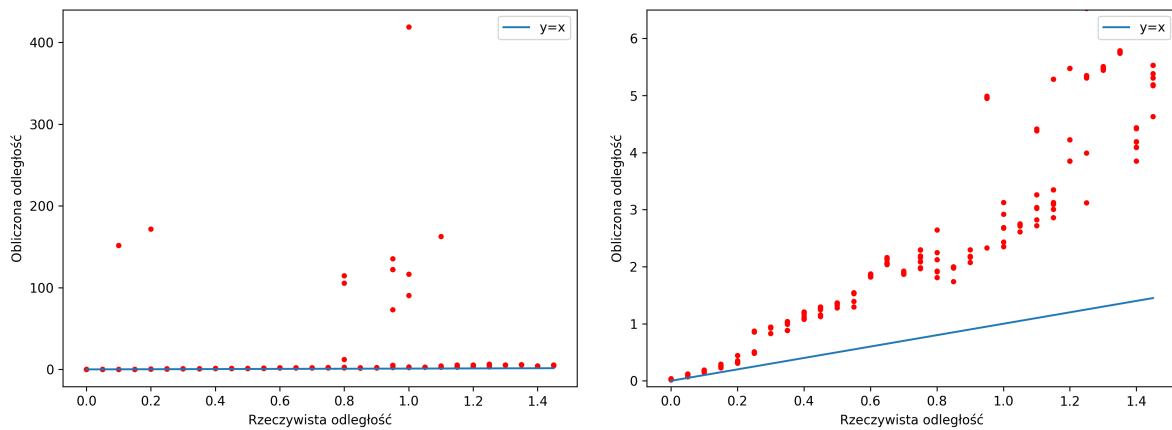
Rys. 5.1: Pomiar obliczanych odległości 1.



Rys. 5.2: Pomiar obliczanych odległości 2.



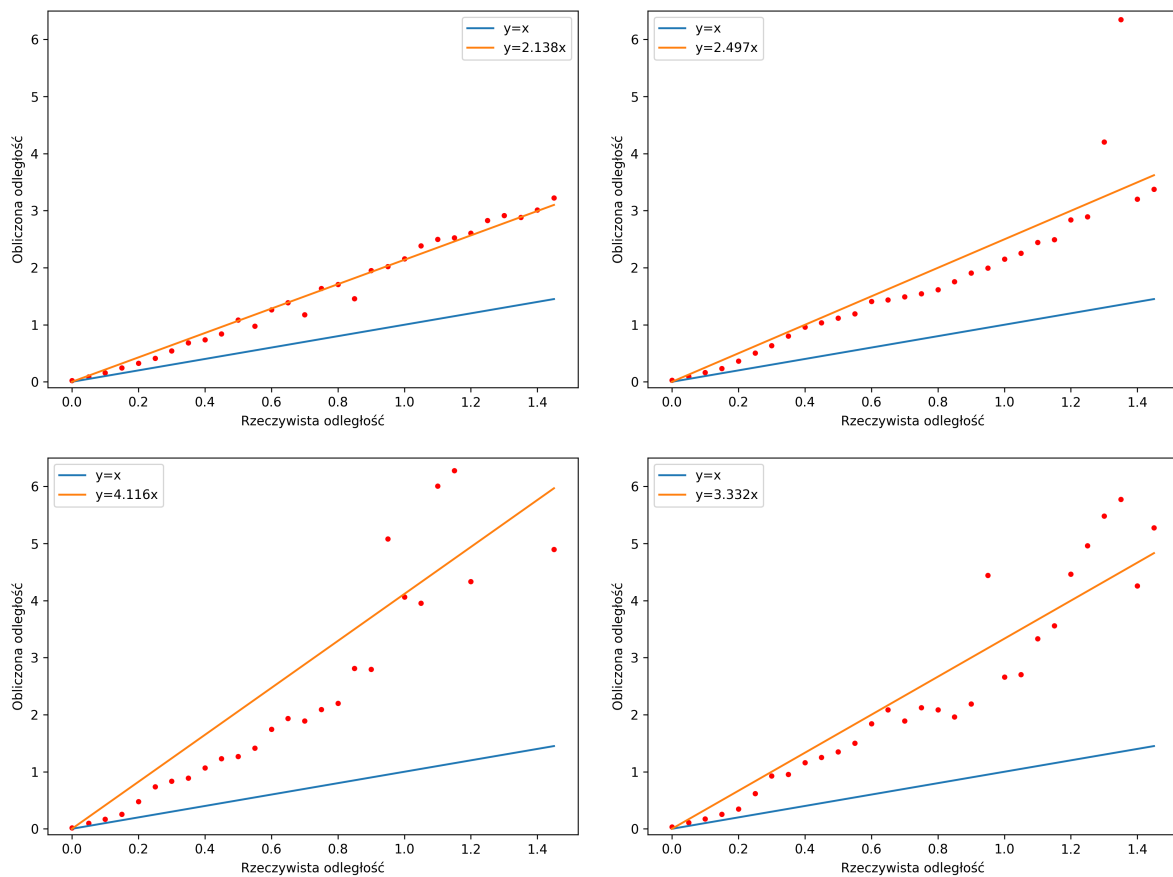
Rys. 5.3: Pomiar obliczanych odległości 3.



Rys. 5.4: Pomiar obliczanych odległości 4.

Aby lepiej odczytać informacje z wykresu uśrednimy pomiary dla każdej z badanych odległości i dodajemy do nich funkcje liniowe o współczynniku otrzymanym przy pomocy regresji liniowej z tychże uśrednionych punktów.

Na wykresach można zauważyć trendy związane ze zmniejszającą się czułością wzmacniacza mikrofonu:



Rys. 5.5: Średnie obliczonych odległości

- zmniejszanie się odległości punktu, w którym czułość jest zbyt mała by niezawodnie wrywać nadawane sygnały,
- współczynnik prostej aproksymującej skalowane odległości rośnie.

Podobne efekty zaobserwowano kiedy brzęczyk nie był kierowany bezpośrednio w kierunku odbiornika.

5.2. Ewaluacja działania systemu

5.3. Wyniki

5.3.1. Interpretacja

5.3.2. Wnioski

Podsumowanie

Literatura

- [1] U. G. P. Office, “govinfo.gov.” <https://www.govinfo.gov/content/pkg/PLAW-106publ81/pdf/PLAW-106publ81.pdf>, 1999. [Accessed 14-05-2024].
- [2] Y. Li, Y. Zhuang, X. Hu, Z. Gao, J. Hu, L. Chen, Z. He, L. Pei, K. Chen, M. Wang, X. Niu, R. Chen, J. Thompson, F. M. Ghannouchi, and N. El-Sheimy, “Toward location-enabled iot (le-iot): Iot positioning techniques, error sources, and error mitigation,” *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4035–4062, 2021.
- [3] A. Norrdine, “An algebraic solution to the multilateration problem,” in *Proceedings of the 15th international conference on indoor positioning and indoor navigation*, Sydney, Australia, vol. 1315, 2012.
- [4] W. Hereman, “Determination of a position in three dimensions using trilateration and approximate distances,” *Colorado School of Mines*, 1995.
- [5] S. Wiszniewski, O. Błaszkiwicz, A. Olejniczak, J. Sadowski, and J. Stefański, “Implementation of the innovative radiolocalization system vcs-mlat (voice communication system – multilateration),” in *2020 Baltic URSI Symposium (URSI)*, pp. 95–99, 2020.
- [6] Z. Yong-xing, Z. Chao, and Y. Yu-lei, “Research on computer time synchronization by software method,” in *2011 International Conference on Electronics, Communications and Control (ICECC)*, pp. 998–1001, 2011.
- [7] D. Mills, “Internet time synchronization: the network time protocol,” *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1482–1493, 1991.