

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

<b>1. Przedstawienie problemu</b>	<b>5</b>
1.1. State of the art	6
<b>2. Sprzęt systemowy</b>	<b>7</b>
2.1. Serwer MQTT	7
2.2. Węzeł	8
2.3. Serwer obliczeniowy	8
<b>3. Eksperyment zerowy</b>	<b>9</b>
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
<b>4. Synchronizacja czasu</b>	<b>14</b>
4.1. Synchronizacja programowa	14
4.1.1. Algorytm synchronizacji NTP	14
4.1.2. Pomiar różnic zegarów	14
4.2. Synchronizacja sprzętowa	14
4.2.1. Synchronizacja z użyciem mikrofonów	14
<b>5. Metody multilateracji</b>	<b>16</b>
5.1. Omówienie zastosowanych metod	16
5.1.1. Układ równań liniowych	16
5.1.2. Liniowa metoda najmniejszej sumy kwadratów	16
5.1.3. Nieliniowa metoda najmniejszej sumy kwadratów	16
5.1.4. Rozkład według wartości szczególnych (SVD)	16
5.2. Wyniki	16
5.2.1. Interpretacja	16
5.2.2. Wnioski	16
<b>6. Podsumowanie</b>	<b>17</b>
<b>Literatura</b>	<b>18</b>

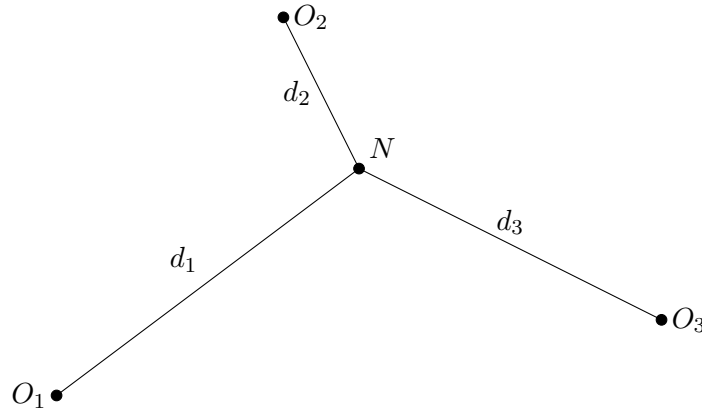
# Skróty

**WASN** (ang. *Wireless Audio Sensor Networks*)

# Rozdział 1

## Przedstawienie problemu

Multilateracja jest techniką lokalizacji pozwalającą obliczyć nieznane koordynaty punktu na podstawie odległości od innych, znanych punktów. Weźmy dwuwymiarowy egzemplarz naszego problemu (Rys. 1.1), gdzie  $N$  - nadajnik,  $O_i$  - odbiorniki,  $d_i$  - odległości



Rys. 1.1: Egzemplarz problemu multilateracji

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

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

który może zostać przekształcony do postaci

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

lub w reprezentacji macierzowej,

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

którą można przedstawić jako

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

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. State of the art

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

- układ równań liniowych[1],
- metoda najmniejszej sumy kwadratów[1],[2],
- nieliniowa metoda najmniejszej sumy kwadratów[1],
- rozkład według wartości szczególnych[1].

Na szczególną uwagę zasługuje praca [1], w której przeprowadzono eksperymenty porównawcze wyżej wymienionych metod.

Metoda	Liczba lokalizacji na 1000 obliczona z dokładności powyżej 5 stóp na podstawie przybliżonych odległości
Układ równań	919
Liniowe najmniejsze kwadraty	856
Nieliniowe najmniejsze kwadraty	81

Tab. 1.1: Porównanie wyników z pracy [1]

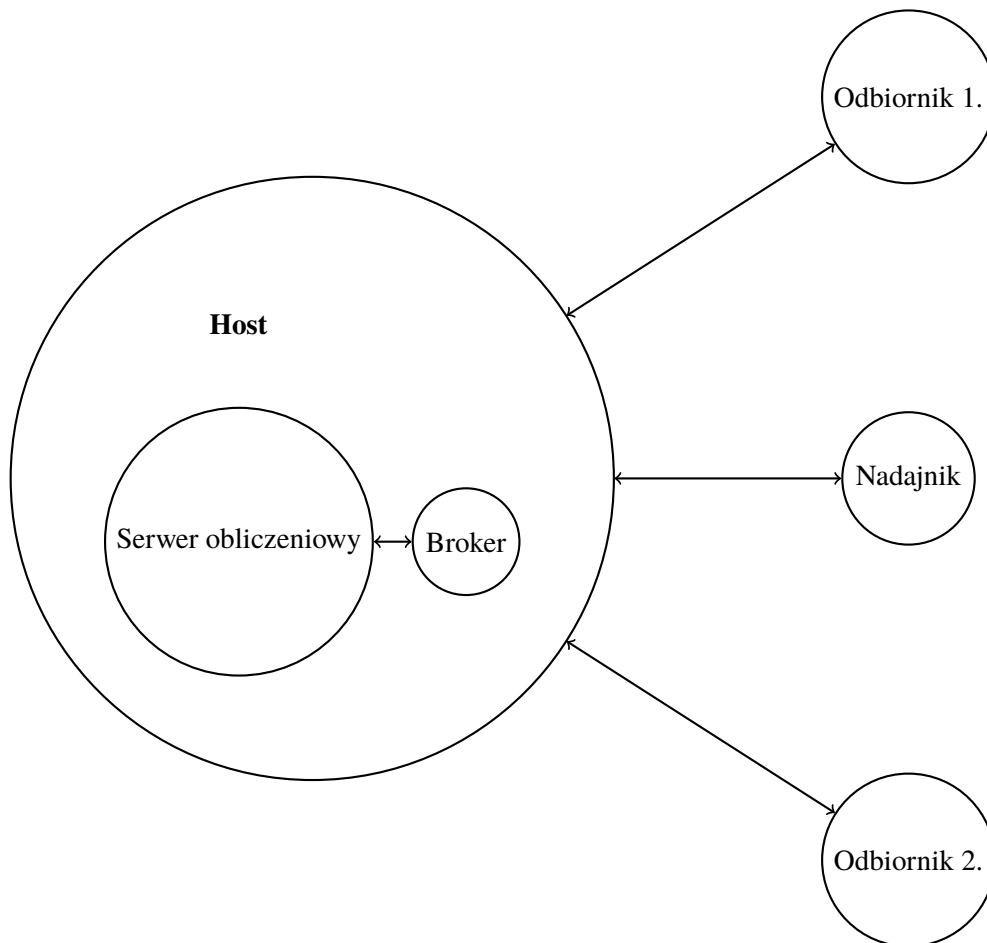
Motywacją tej pracy jest brak dostępnych prac obejmujących to zagadnienie w spektrum fal dźwiękowych, w przeciwieństwie do fal elektromagnetycznych, an których opierają się systemy opisane w pozycjach bibliograficznych. Pytania, które zostały postawione:

-

## Rozdział 2

# Sprzęt systemowy

Znaczna większość prac adresujących temat multilateracji opiera się na systemach urządzeń działających w zakresie fal elektromagnetycznych ([1],[2],[3]). W tej pracy poświęcimy uwagę systemowi działającemu w domenie dźwięku, jak ten aspekt wpływa na skuteczność i dokładność rozwiązywania 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 [2]. Współrzędne odbiorników są znane.

#### 3.1.1. Program węzła

---

**Algorithm 1** Program nadajnika

---

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

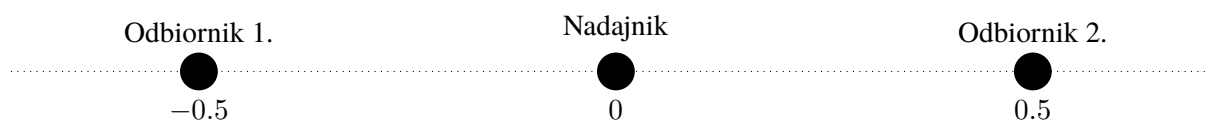
W programie serwera zaimplementowano funkcję rozwiązującą równość 3.1 przy pomocy funkcji bibliotecznych dostępnych w pakiecie *numpy*. Jak łatwo zauważyć pierwsza część lewej strony równości zależy jedynie od macierzy  $A$ , która nie zawiera zmiennych związanych z węzłem źródłowym, co pozwala na wcześniejsze 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.

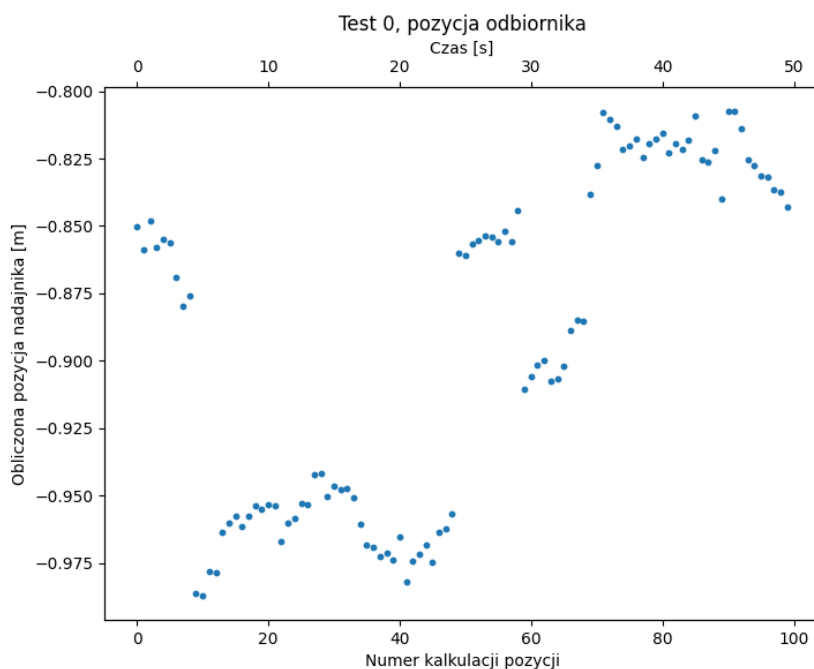
### 3.2.1. Struktura systemu testowego

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

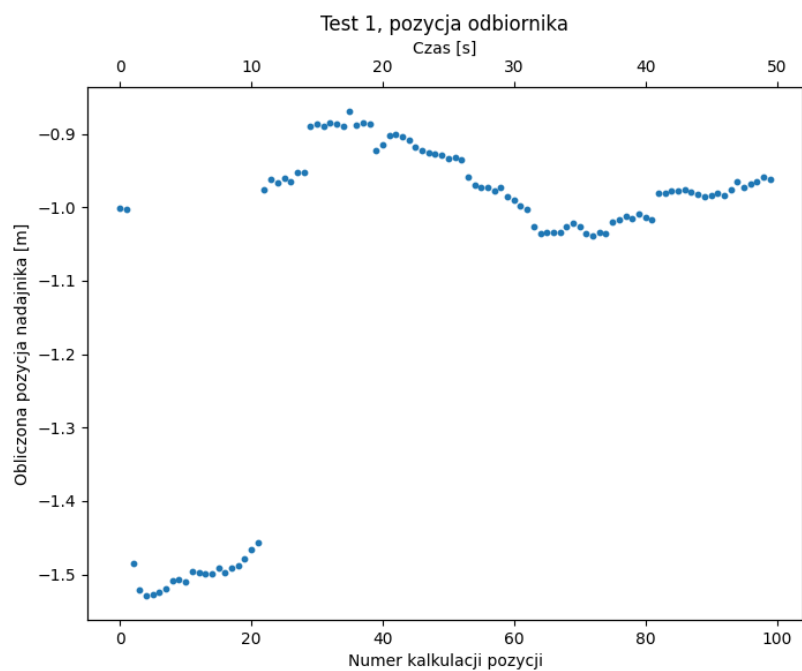
### 3.3. Interpretacja wyników i wnioski

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 odchyłeń jest większa niż odległość między nadajnikami, a dokładność jest absolutnie niezadowalająca. Kolejne testy jedynie utwierdzają te obserwacje.

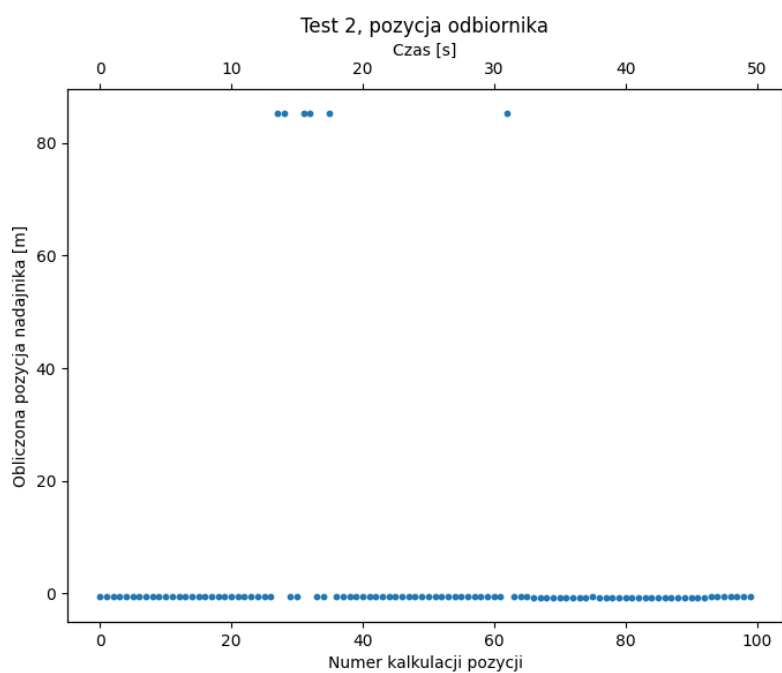
Na niską dokładność i stabilność wyników może mieć wpływ wiele czynników takich jak:

- 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 [4], 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 [5]. 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  $\delta_i$  i przesunięcie zegara  $\theta_i$  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:

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

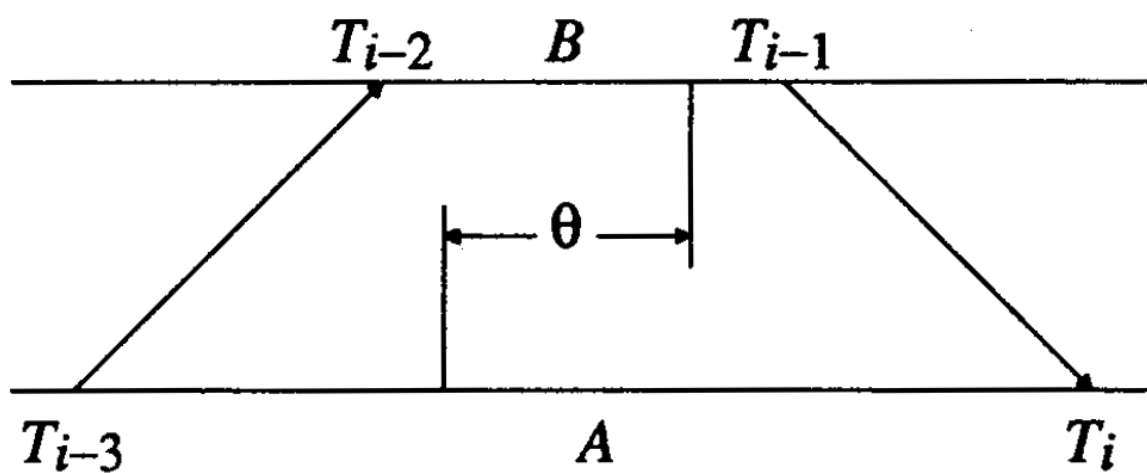
Jak łatwo zauważyć im krótszy jest czas propagacji tym lepsze przybliżenie dostajemy.

Na podstawie tych informacji podjęto próbę synchronizacji programowej zegarów węzłów w systemie multilateracyjnym.

#### 4.1.2. Pomiar różnic zegarów

### 4.2. Synchronizacja sprzętowa

#### 4.2.1. Synchronizacja z użyciem mikrofonów



Rys. 4.1: Pomiar opóźnienia transmisji i przesunięcia zegara

## **Rozdział 5**

# **Metody multilateriacji**

### **5.1. Omówienie zastosowanych metod**

#### **5.1.1. Układ równań liniowych**

#### **5.1.2. Liniowa metoda najmniejszej sumy kwadratów**

#### **5.1.3. Nieliniowa metoda najmniejszej sumy kwadratów**

#### **5.1.4. Rozkład według wartości szczególnych (SVD)**

### **5.2. Wyniki**

#### **5.2.1. Interpretacja**

#### **5.2.2. Wnioski**



## **Rozdział 6**

# **Podsumowanie**

# Literatura

- [1] W. Hereman, “Determination of a position in three dimensions using trilateration and approximate distances,” *Colorado School of Mines*, 1995.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] D. Mills, “Internet time synchronization: the network time protocol,” *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1482–1493, 1991.