

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	8
1.1. State of the art	9
2. Sprzęt systemowy	10
2.1. Serwer MQTT	10
2.2. Węzeł	11
2.3. Serwer obliczeniowy	11
3. Eksperyment zerowy	12
3.1. Opis działania	12
3.1.1. Program węzła	12
3.1.2. Program serwera	14
3.1.3. Opis algorytmu	14
3.2. Ewaluacja działania systemu	14
3.2.1. Struktura systemu testowego	14
3.3. Interpretacja wyników i wnioski	15
4. Synchronizacja czasu	16
4.1. Synchronizacja programowa	16
4.1.1. Algorytm synchronizacji NTP	16
4.1.2. Pomiar różnic zegarów	16
4.2. Synchronizacja sprzętowa	16
4.2.1. Synchronizacja z użyciem mikrofonów	16
5. Metody multilateracji	17
5.1. Omówienie zastosowanych metod	17
5.1.1. Układ równań liniowych	17
5.1.2. Liniowa metoda najmniejszej sumy kwadratów	17
5.1.3. Nieliniowa metoda najmniejszej sumy kwadratów	17
5.1.4. Rozkład według wartości szczególnych (SVD)	17
5.2. Wyniki	17
5.2.1. Interpretacja	17
5.2.2. Wnioski	17
6. Podsumowanie	18
Literatura	19

Spis rysunków

1.1. Egzemplarz problemu multilateracji	8
2.1. Topologia systemu	10
3.1. Układ systemu testowego	14

Spis tabel

1.1. Porównanie wyników z pracy [1]	9
---	---

Spis listingów

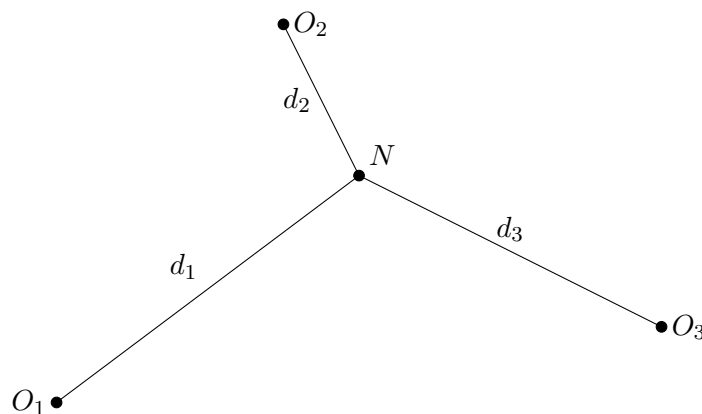
Skróty

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

$$A \cdot x = b \quad (1.4)$$

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]

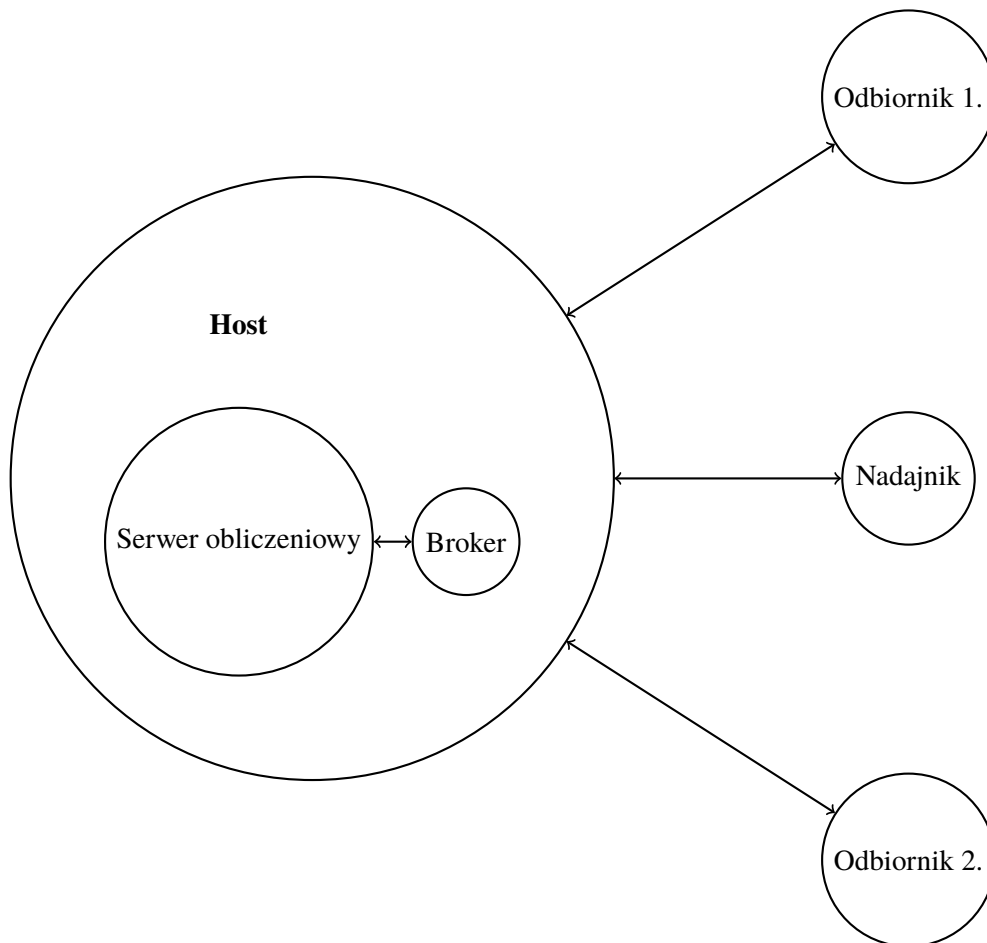
Motywacją tej pracy jest brak dostępnych prac obejmujących to zagadnienie w spektrum fal dźwiękowych, w przeciwieństwie do fal elektromagnetycznych, an których opierają się systemy opisane w pozycjach bibliograficznych. Pytania, które zostały postawione:

-

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].

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,
- *TOPIC* - nazwa kanału MQTT korespondującego do danego węzła,

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
```

- *TIME_TOPIC* - nazwa kanału MQTT służącego do przesyłania wiadomości o synchronizacji czasu.

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

```

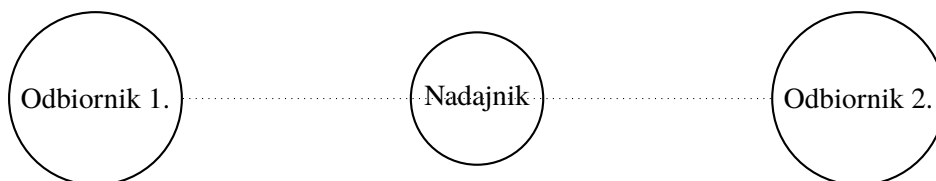
- A - macierz zawierająca współrzędne obrotów z równania 1.4
- SS - prędkość dźwięku $\left[\frac{m}{s}\right]$,
- $\text{time}()$ - czas dostarczony w ostatniej wiadomości od węzła,
- $\text{coords}()$ - współrzędne węzła,
- t_1 - czas pomiędzy obliczeniami pozycji,
- t_2 - czas pomiędzy wiadomościami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

3.2. Ewaluacja działania systemu

3.2.1. Struktura systemu testowego



Rys. 3.1: Układ systemu testowego

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.

3.3. Interpretacja wyników i wnioski

Rozdział 4

Synchronizacja czasu

4.1. Synchronizacja programowa

4.1.1. Algorytm synchronizacji NTP

4.1.2. Pomiar różnic zegarów

4.2. Synchronizacja sprzętowa

4.2.1. Synchronizacja z użyciem mikrofonów

Rozdział 5

Metody multilateracji

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.