

# Bežične mreže osjetila

## Iterativna lokalizacija - trilateracija

Dario Barać, Mario Maričević  
Diplomski sveučilišni studij računarstva, RITEH

Svibanj, 2022

### Sadržaj

<b>1</b>	<b>Opis problema</b>	<b>2</b>
1.1	Ulazi u algoritam . . . . .	2
1.2	Ograničenja . . . . .	2
<b>2</b>	<b>Odabrani pristup</b>	<b>2</b>
2.1	Opis algoritma . . . . .	3
2.2	Statusi . . . . .	3
2.3	Poruke . . . . .	3
2.4	Sadržaj memorije čvorova . . . . .	4
2.5	Pseudokod . . . . .	5
<b>3</b>	<b>Analiza vremenske i komunikacijske složenosti - TODO</b>	<b>8</b>

# 1 Opis problema

Za danu bežičnu mrežu osjetila definiranu skupom čvorova u 2D prostoru, vezama između čvorova i udaljenostima između čvorova koji su povezani, potrebno je pomoću postupka trilateracije odrediti položaj svih čvorova u relativnom koordinatnom sustavu sa čvorom inicijatorom u ishodištu. Pretpostavlja se da šum nije prisutan kod mjerenja udaljenosti i da je formacija grafa globalno kruta, odnosno da je moguće jednoznačno odrediti položaje svih čvorova u danom koordinatnom sustavu.

## 1.1 Ulazi u algoritam

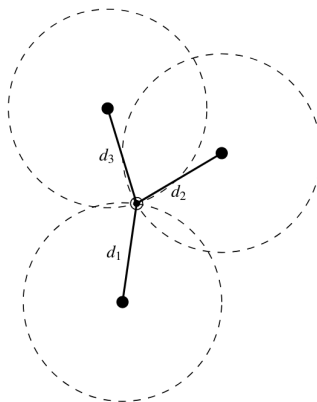
- Skup čvorova i veza
- Udaljenost između svakog para povezanih čvorova

## 1.2 Ograničenja

- Jedinstveni inicijator
- Dvosmjerna komunikacija između povezanih čvorova
- Apsolutna pouzdanost
- Globalno kruta formacija ulaznog grafa

# 2 Odabrani pristup

U radu [1] opisani su razni pristupi za mjerenje udaljenosti među čvorova i algoritmi za lokalizaciju. Za određivanje položaja trilateracijom (slika 1) potrebno je znati udaljenosti od čvorova koji znaju svoj položaj do čvora čiji položaj nije poznat. Neki od načina procjene udaljenosti su pomoću snage dobivenog signala (*Received Signal Strength - RSS*) ili vremena propagacije signala (*ToA, TDoA*).



Slika 1: Određivanje položaja trilateracijom [1]

Metode za lokalizaciju mogu se podijeliti na centralizirane i distribuirane. Kod centraliziranih metoda svi čvorovi udaljenosti do svojih susjeda prosljeđuju glavnom čvoru, koji određuje položaj svih ostalih čvorova. Kod distribuiranih metoda čvorovi sami određuju svoj položaj pomoću informacija dobivenih od susjeda. Prednost centraliziranih metoda je to da se smanjuju zahtjevi

(procesorska moć i memorija) svih osim glavnog čvora, ali u odnosu na distribuirane metode zahtijevaju značajno veći broj poslanih poruka i manje su pouzdane (prestanak rada glavnog čvora onemogućuje lokalizaciju). Zbog toga je odabran distribuirani pristup lokalizaciji.

## 2.1 Opis algoritma

Pristup koji je odabran kao rješenje problema sastoji se od dvije faze:

### 1. Određivanje inicijalnog krutog segmenta:

- odabere se inicijator prije pokretanja algoritma npr. sa ID=0
- inicijator pronađe dva susjeda koji su osim s njim povezani i međusobno (listu svih svojih susjeda šalje svim susjedima, koji odgovaraju odgovaraju sa listama zajedničkih susjeda i udaljenostima do njih)
- inicijator koristi znanje o udaljenostima (do svoja dva susjeda i između njih) za izračun inicijalnog krutog segmenta - trokuta (Assumption Based Coordinate (ABC) algorithm [2]):
  - sebe stavi u ishodište (0,0)
  - prvog susjeda stavi na  $x$  os ( $distance_{01}$ , 0)
  - za drugog susjeda su moguća 2 položaja, odabere se položaj sa pozitivnom  $y$  vrijednosti
- pomoću znanja o zajedničkim susjedima i udaljenostima do njih, inicijator u petlji trilateracijom dodaje u inicijalni kruti segment susjede za koje je to moguće (oni koji imaju barem 2 zajednička susjeda s inicijatorom, koji već jesu u krutom segmentu)
- inicijator pošalje tim susjedima njihov položaj

### 2. Dodavanje ostalih čvorova u kruti segment (lokalizacija preostalih čvorova):

- inicijator i susjedi iz prve faze šalju svoj položaj svim svojim susjedima
- ostali čvorovi računaju svoj položaj kada saznaju položaj od tri susjeda i šalju ga svojim susjedima čiji položaj ne znaju

## 2.2 Statusi

- **INITIATOR**, čvor koji definira inicijalni kruti segment lokalizirane mreže i određuje položaj svojih susjeda koji su u inicijalnom segmentu
- **WAITING\_FOR\_FIX**, inicijalno stanje svim čvorovima osim inicijatoru, označava da su potrebne dodatne informacije kako bi čvor mogao odrediti svoj položaj
- **LOCALIZED**, čvor koji zna svoj položaj - algoritam je završen kada svi čvorovi u mreži imaju ovaj status

## 2.3 Poruke

- *CommonNeighborQuery* - sadrži popis svih susjeda inicijatora, inicijator šalje ovu poruku svojim susjedima

- *CommonNeighborResponse* - odgovor susjeda inicijatoru, *None* ako nemaju zajedničkog susjeda ili (*neighbors*, *distances*) ako imaju zajedničkog susjeda (*neighbors* je lista zajedničkih susjeda, *distances* je lista udaljenosti do njih)
- *OwnPosition* - sadrži (*x,y*), položaj čvora primatelja, inicijator šalje ovu poruku čvorovima koji su dio inicijalnog krutog segmenta mreže
- *NeighborPosition* - sadrži (*x,y*), položaj susjeda koji šalje poruku

## 2.4 Sadržaj memorije čvorova

- *neighbors* - lista susjeda
- *neighborDistances* - lista udaljenosti do svih susjeda čvora
- *neighborPositions* - lista duljine jednake kao lista susjeda, svaki element je *None* ili (*x,y*), ovisno o tome zna li čvor položaj tog susjeda
- *nKnownNeighborPositions* - broj susjeda za koje čvor zna položaj
- *position* - sadrži (*x,y*), položaj čvora
- *unvisitedNeighbors* - lista koju čvor inicijator koristi kod konstrukcije početnog krutog segmenta formacije mreže, sadrži sve čvorove koji još nisu odgovorili na *CommonNeighborQuery* i inicijalno je jednaka listi svih susjeda
- *rigidSegment* - skup elemenata tipa (*node\_id*, *node\_pos*) u memoriji inicijatora, sadrži susjede inicijatora koji su dio inicijalnog krutog segmenta
- *commonNeighborLists* - skup parova (*node\_id*, *common*) u memoriji inicijatora, *node\_id* je ID susjeda, *common* je lista susjeda zajedničkih inicijatoru i tom susjedu. Svaki element liste *common* sadrži ID susjeda i udaljenost do njega

## 2.5 Pseudokod

---

Distribuirani algoritam za iterativnu lokalizaciju (1) - status INITIATOR

---

INITIATOR

*Spontaneously*

**begin**

**send**(CommonNeighborQuery{*neighbors*}) **to** *neighbors*;

**end**

*Receiving*(CommonNeighborResponse)

**begin**

$unvisitedNeighbors \leftarrow unvisitedNeighbors - sender$ ;

**if** commonNeighborResponse  $\neq$  None **then**

$common \leftarrow commonNeighborResponse$ ;

$commonNeighborLists \leftarrow commonNeighborLists + (sender, common)$ ;

**end if**

**if**  $unvisitedNeighbors = \emptyset$  **then**

$n_1, common_1 \leftarrow x \in_R commonNeighborLists$ ; /\* any neighbor with common neighbors \*/

$dist_{01} \leftarrow \mathbf{dist}(n_1)$ ;

$n_2, dist_{12} \leftarrow x \in_R common_1$ ; /\* any common neighbor \*/

$dist_{02} \leftarrow \mathbf{dist}(n_2)$ ;

$rigidSegment \leftarrow \mathbf{defineInitialRigidSegment}(dist_{01}, dist_{02}, dist_{12}, n_1, n_2)$ ;

$commonNeighborLists -= \{(n_1, common_1), (n_2, common_2)\}$ ;

**addNeighborsToRigidSegment**( $rigidSegment, commonNeighborLists$ );

**for** ( $neigh, pos$ ) **in**  $rigidSegment$  **do**

**send**(OwnPosition{ $pos$ }) **to**  $neigh$ ;

**end for**

**send**(NeighborPosition{ $position$ }) **to**  $neighbors - rigidSegment_{ids}$ ;

        become LOCALIZED;

**end if**

**end**

---

WAITING\_FOR\_FIX

*Receiving*(CommonNeighborQuery)

**begin**

*initiatorNeighbors*  $\leftarrow$  CommonNeighborQuery;

*allCommon*  $\leftarrow$  *neighbors*  $\cap$  *initiatorNeighbors*;

**if** *allCommon*  $\neq \emptyset$  **then**

**send**(CommonNeighborResponse{*allCommon*, **dist**(*allCommon*)) **to** sender;

**else**

**send**(CommonNeighborResponse{*None*}) **to** sender;

**end if**

**end**

*Receiving*(OwnPosition)

**begin**

*position*  $\leftarrow$  OwnPosition;

**send**(NeighborPosition{*position*}) **to** *neighbors* - sender;

become LOCALIZED;

**end**

*Receiving*(NeighborPosition)

**begin**

*nPos*  $\leftarrow$  NeighborPosition;

*neighborPositions*[*sender*]  $\leftarrow$  *nPos*;

*nKnownNeighborPositions* + = 1;

**if** *nKnownNeighborPositions* = 3 **then**

*position*  $\leftarrow$  **trilaterate**(*neighborPositions*, *neighborDistances*);

*knownNeighbors*  $\leftarrow$  {*n*  $\in$  *neighbors* | *neighborPositions*[*n*]  $\neq$  *None*};

**send**(NeighborPosition{*position*}) **to** *neighbors* - *knownNeighbors*;

become LOCALIZED;

**end if**

**end**

---

```

procedure defineInitialRigidSegment( $r_{01}, r_{02}, r_{12}, neigh_1, neigh_2$ )
  begin
     $position \leftarrow (0, 0);$  /* set initiator position */;
     $x_1, y_1 \leftarrow (r_{01}, 0);$ 
     $x_2 \leftarrow \frac{r_{01}^2 + r_{02}^2 + r_{12}^2}{2r_{01}};$ 
     $y_2 \leftarrow \sqrt{r_{02}^2 - x_2^2};$ 
     $rigidSegment \leftarrow \{(neigh_1, (x_1, y_1)), (neigh_2, (x_2, y_2))\};$ 
    return  $rigidSegment;$ 
  end

procedure trilaterate(( $x_1, y_1$ ), ( $x_2, y_2$ ), ( $x_3, y_3$ ), ( $r_1, r_2, r_3$ ))
  begin
     $A \leftarrow -2x_1 + 2x_2;$ 
     $B \leftarrow -2y_1 + 2y_2;$ 
     $C \leftarrow r_1^2 - r_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2;$ 
     $D \leftarrow -2x_2 + 2x_3;$ 
     $E \leftarrow -2y_2 + 2y_3;$ 
     $F \leftarrow r_2^2 - r_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2;$ 
     $x \leftarrow \frac{CE - FB}{EA - BF};$ 
     $y \leftarrow \frac{C - Ax}{B};$ 
    return ( $x, y$ );
  end

procedure addNeighborsToRigidSegment( $rigidSegment, commonNeighborLists$ )
  begin
     $rigidSegmentUpdated \leftarrow true;$ 
    repeat
       $rigidSegmentUpdated \leftarrow false;$ 
      for ( $neigh, common$ ) in  $commonNeighborLists$  do
         $rigidCommon \leftarrow \{(n_{id}, n_{dist}) \in common | n_{id} \in rigidSegment\};$ 
        if  $|rigidCommon| \geq 2$  do
           $ids \leftarrow \{n_{id} | n \in rigidCommon\};$ 
           $positions \leftarrow \{n_{pos} \in rigidSegment | n_{id} \in ids\} + (0, 0);$ 
           $distances \leftarrow \{n_{dist} | n \in rigidCommon\} + \mathbf{dist}(neigh);$ 
           $pos \leftarrow \mathbf{trilaterate}(positions, distances);$ 
           $rigidSegment \leftarrow rigidSegment + (neigh, pos);$ 
           $commonNeighborLists -= (neigh, common);$ 
           $rigidSegmentUpdated \leftarrow true;$ 
        endif
      end for
    until  $rigidSegmentUpdate = false$ 
  end

```

---

### **3 Analiza vremenske i komunikacijske složenosti - TODO**



## Literatura

- [1] Roudy Dagher, Roberto Quilez. Localization in Wireless Sensor Networks. Nathalie Mitton and David Simplot-Ryl. Wireless Sensor and Robot Networks From Topology Control to Communication Aspects, Worldscientific, pp.203-247, 2014, 978-981-4551-33-5. [⟨10.1142/9789814551342\\_0009⟩](#). [⟨hal-00926928⟩](#)
- [2] Savarese, C., Rabaey, J. M., Beutel, J. (n.d.). Location in distributed ad-hoc wireless sensor networks. 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221). doi:10.1109/icassp.2001.940391