Technical Report

Mathias Pelka*

Position Calculation with Least Squares based on Distance Measurements

Abstract: Position estimation based on distances is a well understood problem. This document describes a simple way to linearize the position equation. Based on the linearization the problem is solved step by step using least squares. This paper includes an example implementation in Matlab.

1 Introduction

Localization in general plays a major role in emergent applications in the field of medical, industrial or consumer application. Such applications aim for use in indoor application where in general Global Navigation Satellite System (GNSS), e.g. GPS, is not available. This is because the GNSS signal is too weak to penetrate concrete walls or is otherwise shadowed or reflected. Therefore other solutions to determine the position within buildings is required.

In this paper it is assumed that distances between an anchor (or reference node) and a tag is available. The situation is depicted in Figure 1. In this figure several anchors (denoted as A_i) are shown and further a tag is present. Between those anchors and tags distance measurements are available. This is paper does not describe how to obtain such distances but focuses to solve the localization problem. This method can also applied to other similar problems. For details, the reader is refereed to [1].

2 Linearisation

The basic equation, for three dimensions, to describe the localization problem based on lateration is

$$(x-x_i)^2+(y-y_i)^2+(z-z_i)^2=d_i^2. \tag{1}$$

In this equation d_i denotes the ith distance between tag and an anchor [2], with a total of n equations. $\mathbf{r}=[x \quad y]^T$ describes the coordinates which are to be determined. The coordinates

***Corresponding Author: Mathias Pelka:** Department of Electrical Engineering and Computer Science - CoSA Center of Excellence, Lübeck University of Applied Sciences, Universität zu Lübeck, Germany, Graduate School for Computing in Medicine and Life Sciences, e-mail: mathias.pelka@fh-luebeck.de

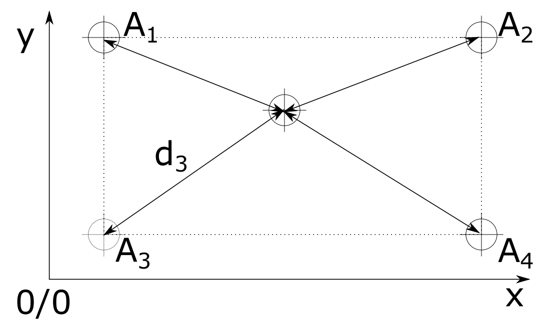

Figure 1: Illustration of localization problem. There are four anchors and a tag present. The distances between tag and anchors (d_i) are known.

$\mathbf{r}_i$ describe the anchor coordinates in Cartesian coordinates. There are n distance measurements for n anchors available.

To linearize Eq. (1) a reference node needs to be chosen. All other equations are subtracted from equation which incorporates the reference anchor. For this derivation the first anchor is chosen as the reference anchor. However, there other linearization techniques are possible. A linearization is found if all other equations are subtracted from the reference equation.

$$\begin{aligned}&\left[(x-x_1)^2+(y-y_1)^2+(z-z_1)^2\right]-\\&\left[(x-x_2)^2+(y-y_1)^2+(z-z_1)^2\right]\\&=d_1^2-d_2^2.\end{aligned} \tag{2}$$

First, the expressions in the parenthesis are squared which results in:

$$\begin{aligned}&\left[x^2-2xx_1+x_1^2+y^2-2yy_1+y_1^2+z^2-2zz_1+z_1^2\right]-\\&\left[x^2-2xx_2+x_2^2+y^2-2yy_2+y_2^2+z^2-2zz_2+z_2^2\right]\\&=d_1^2-d_2^2.\end{aligned} \tag{3}$$

Second, the remaining parentheses are solved:

$$\begin{aligned}&x^2-2xx_1+x_1^2+y^2-2yy_1+y_1^2+z^2-2zz_1+z_1^2-\\&x^2+2xx_2-x_2^2-y^2+2yy_2-y_2^2-z^2+2zz_2-z_2^2\\&=d_1^2-d_2^2.\end{aligned} \tag{4}$$

Now, the squared unknown coordinates $\mathbf{r}$ vanish which results in:

$$\begin{aligned}&-2xx_1-2yy_1-2zz_2+2xx_2+2yy_2+2zz_1+\\&x_1^2+y_1^2+z_1^2-x_2^2-y_2^2-z_2^2=d_1^2-d_2^2.\end{aligned} \tag{5}$$

In the last steps, the equation is rearranged. Everything which is known, e.g. anchor coordinates $\mathbf{r}_i$ are grouped together with the measurements d_i on the right hand side. On the left hand side of the equation the unknown tag coordinates $\mathbf{r}$ are grouped.

$$\begin{aligned} &2x\left(x_2-x_1\right)-2y\left(y_2-y_1\right)+2z\left(z_2-z_1\right) \\ &=d_1^2-d_2^2-x_1^2-y_1^2-z_1^2+x_2^2+y_2^2+z_2^2. \end{aligned} \tag{6}$$

To simplify writing, a new symbol is introduced: $k_i = x_i^2+y_i^2+z_i^2$. This allows for a more compact writing.

$$2x\left(x_2-x_1\right)-2y\left(y_2-y_1\right)-2z\left(z_2-z_1\right)=d_1^2-d_2^2-k_1+k_2. \tag{7}$$

If this subtraction is written more general and one denotes the reference anchor as l and one subtracts the ith equation, one finds:

$$2x\left(x_i-x_l\right)-2y\left(y_i-y_l\right)-2z\left(z_i-z_l\right)=d_l^2-d_l^2-k_l+k_i. \tag{8}$$

For four anchors, as depicted in Figure 1, a system of linear equation is found. This system of equations are written in matrix notation:

$$2\underbrace{\begin{pmatrix} x_2-x_1 & y_2-y_1 & z_2-z_1 \\ x_3-x_1 & y_3-y_1 & z_3-z_1 \\ x_4-x_1 & y_4-y_1 & z_4-z_1 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} x \\ y \\ z \end{pmatrix}}_{\mathbf{r}} = \underbrace{\begin{pmatrix} d_1^2-d_2^2-k_1+k_2 \\ d_1^2-d_3^2-k_1+k_3 \\ d_1^2-d_4^2-k_1+k_4. \end{pmatrix}}_{\mathbf{b}} \tag{9}$$

This is solved using linear algebra:

$$\mathbf{r}=\mathbf{A}^{-1}\mathbf{b}. \tag{10}$$

Are more measurements than unknowns available a solution based on least squares is possible [3].

$$\mathbf{r}=\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b} \tag{11}$$

In this equation $[\cdot]^T$ denotes matrix transpose.

With these equations at hand, it is possible to calculate the position of a tag with respect to known anchor coordinates.

3 Example

The following code shows a Matlab example how to solve a position based on a set of anchor coordinates and given distances.

Listing 1: Algorithm

```
%%clear up
clear all; close all;

%% define coordinates
anchor_matrix=[0 0;10 0;0 10;10 10];
r = [5 5];

%% determine distances
for i = 1:length(anchor_matrix)
    vec = anchor_matrix(i,:)-r;
    d(i) = sqrt(sum(vec.^2))
end

%%generate matrix
x=anchor_matrix(:,1);
y=anchor_matrix(:,2);
k=x.^2 + y.^2;

for i=2:length(d)
    A(i-1,:)=[x(i) y(i)]-[x(1) y(1)];
    b(i-1)=d(1)^2-d(i)^2+k(i)-k(1);
end

%% compute position
pinv(A)*b'/2
```

First, the anchor coordinates together with the tag position r is provided. Second, the distances between the tag and the anchor are calculated. Since this is a symmetrical problem, and the tag is in the center of the anchors, the distances are all the same.

Listing 2: Output

```
>> d
d =
    7.0711    7.0711    7.0711    7.0711
```

In the next step, the algorithm generates the matrix $\mathbf{A}$

Listing 3: Output

```
>> A
A =
    10     0
     0    10
    10    10
```

And last, the algorithm generates $\mathbf{b}$.

Listing 4: Output

```
b =
   100   100   200
```

In the end, a least squares solution is computed with:

Listing 5: Output

```
>> pinv(A)*b'/2
ans =
    5.0000    5.0000
```

With this algorithm it is possible to calculate an arbitrary position provided by $\mathbf{r}$. A natural extension is to investigate the behavior of Least Squares in the presence of noise.

4 Conclusion

This paper showed a simple linearization of the position equation. Further, it was shown how this linearization is used to reformulate the problem into matrix form. This matrix was solved with Least Squares. Last, an example implementation in Matlab along with sample code was presented.

Acknowledgments

This publication is a result of the research within the Center of Excellence CoSA in the m:flo project which is funded by German Federal Ministry for Economic Affairs and Energy (BMWi, FKZ KF3177201ED3).

References

[1] M. Laaraiedh, L. Yu, S. Avrillon, and B. Uguen, "Comparison of hybrid localization schemes using rssi, toa, and tdoa," in *Wireless Conference 2011-Sustainable Wireless Technologies (European Wireless), 11th European*. VDE, 2011, pp. 1–5.

[2] M. Pelka, C. Bollmeyer, and H. Hellbrück, "Indoor Localization based on Bi-Phase Measurements for Wireless Sensor Networks," in *2015 IEEE Wireless Communications and Networking Conference (WCNC): - Track 3: Mobile and Wireless Networks (IEEE WCNC 2015 - Track 3- Mobile and Wireless Networks)*, New Orleans, USA, Mar. 2015.

[3] Z. Lifang, M. Pelka, C. Bollmeyer, and H. Hellbrück, "Comparison and Performance Evaluation of Indoor Localization Algorithms based on an Error Model for an Optical System," March 2015.