RSS Ranging Method Implementations

Kerem Karataş

April 7, 2015

Contents

1 Introduction

In [1] three methods of RSS ranging are discussed and tested. This document is prepared to show the outputs of the implementations in Matlab. Under Section 2 the theory behind is briefly explained with partial Matlab codes that implement related parts.

2 Implementation of RSS Ranging Techniques

During implementation, each technique is tested with Monte Carlo placement of sensors and emitters under no additional error sources introduced to the system. The demonstrated outputs in the following sections are all using the same placements as shown below;

- Region of Interest is: 100m x 100m
- Region is divided into grids of size: 5m x 5m
- Number of sensors: 4
- Path Loss Exponent (Actual): 3.5
- Path Loss Exponent (Assumed): 3.5
- Transmit Power (Actual): 1 Watt
- Transmit Power (Assumed): 1 Watt
- Sensor 1 is located at (82.5314,8.347)
- Sensor 2 is located at (13.3171,17.3389)
- Sensor 3 is located at (39.0938,83.138)
- Sensor 4 is located at (39.9258,52.6876)
- Emitter is placed at (41.6799,65.686)
- Shadow Spread (dB): 0

The area is separated into grids to make sure that emitters and sensors are not placed in the same grid. Because the aim of the simulations is to show implementations are working properly additional noise is not added such as shadow spread or misinformation about Transmit Power/Path Loss Exponent.

2.1 Min-Max

Distances are estimated from each sensor. A square is formed using sensor location as the center and sides twice the length of the estimated distance. Estimation of the emitter is done as the center of the quadrilateral formed by the intersections of the squares. Using analytic geometry diagonally opposing centers of this quadrilateral is given in Equation 1 where (x_i, y_i) are coordinates of i^{th} sensor and d_i is distance of the emitter to the i^{th} sensor.

$$(max(x_i - d_i), max(y_i - d_i)) \times (min(x_i + d_i), min(y_i + d_i)) \tag{1}$$

Implementation of this is done with the following lines in Matlab;

```
% Estimate the distance from sensors to the emitter using FSPL
estDist = (r./P_E).^(-1/alpha_assumed);

% Create a square by finding the intersection of circles that centers known
% sensor locations with its related estimated distance.
xLine = [max(sPos(1,:)-estDist) min(sPos(1,:)+estDist)];
yLine = [max(sPos(2,:)-estDist) min(sPos(2,:)+estDist)];

% Find the center of the square. Estimated location of emitter is here.
ex = mean(xLine);
ey = mean(yLine);
```

Simulation outputs are;

- Emitter is found at (39.9258,65.6496)
- Estimation Error is 1.7545 meters

Placement of Sensors/Emitter and estimated emitter location is shown in Figure 1. It is hard to see the quadrilateral in that figure therefore a zoomed in version is provided in Figure 2.

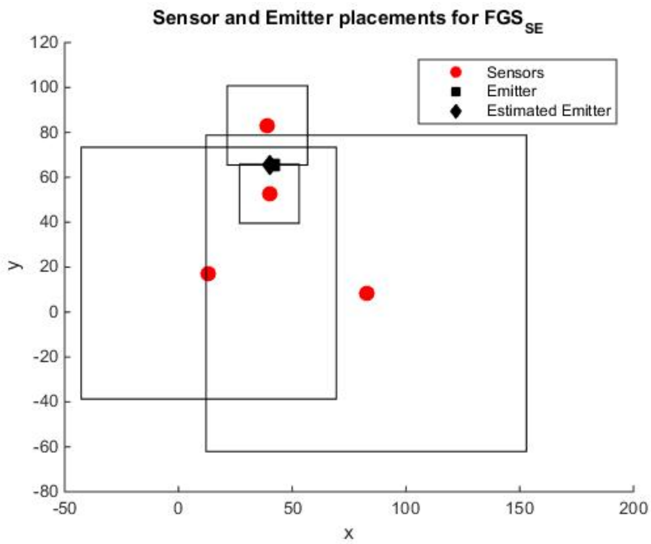

Figure 1: Placement of sensors and emitter in the region of interest and the estimation location using Min-Max with grid length 5, region of interest 100m x 100m, path loss exponent 3.5 and no additional noise. Estimation error is 1.7545 meters, emitter is found at (39.9258,65.6496).

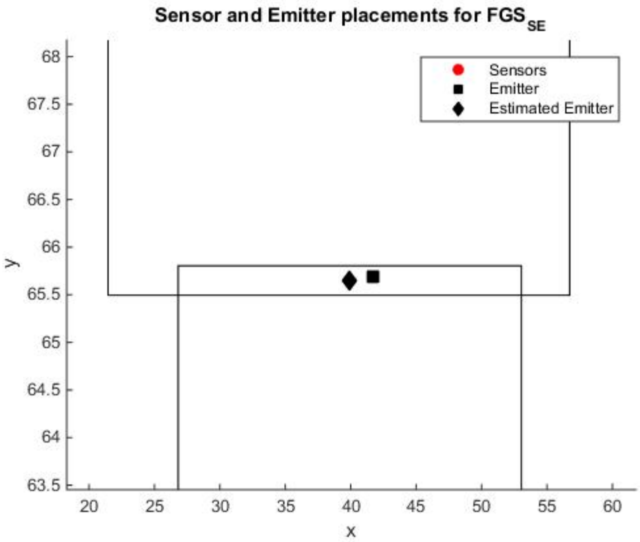

Figure 2: Zoomed in version of Figure 1 towards the actual emitter location and estimated emitter location.

2.2 Maximum Likelihood

Error shown in Equation 2 is minimized.

$$e_i(x_0, y_0) = d_i - \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \tag{2}$$

Minimizing the error is done by solving the equation in the form $y = Xb$ where y and X are shown in Equation 3 and 4. In these equations x_k, y_k and d_k are k$^{\text{th}}$ sensors x coordinate, y coordinate and estimated distance from the emitter.

$$y = \begin{bmatrix} -x_1^2 - y_1^2 + d_1^2 - (-x_k^2 - y_k^2 + d_k^2) \\ \vdots \\ -x_{k-1}^2 - y_{k-1}^2 + d_{k-1}^2 - (-x_k^2 - y_k^2 + d_k^2) \end{bmatrix} \tag{3}$$

$$X = \begin{bmatrix} 2(x_k - x_1) & 2(y_k - y_1) \\ \vdots & \vdots \\ 2(x_k - x_{k-1}) & 2(y_k - y_{k-1}) \end{bmatrix} \tag{4}$$

Implementation of this technique is done using the following partial code;

```
% Find estimated distance from all sensors
estDist = (r./P_E).^(-1/alpha_assumed);

% Creation of $X = [2(x_{N_s} - x_a) 2(y_{N_s} - y_a)]$ where a is all
% numbers from 1 to $N_s-1$.
X = zeros(N_s-1,2);
for k = 1:N_s-1
    X(k,1)=2*(sPos(1,N_s)-sPos(1,k));
    X(k,2)=2*(sPos(2,N_s)-sPos(2,k));
end

% Creation of;
% $y = [-{x_a}^2-{y_a}^2+{d_a}^2-(-{x_{N_s}}^2-{y_{N_s}}^2+{d_{N_s}}^2)]$
```

```
% where a is all numbers from 1 to $N_s-1$.
y = zeros(N_s-1,1);
for k = 1:N_s-1
    y(k)= (-sPos(1,k)^2-sPos(2,k)^2+estDist(k)^2 ...
        -(-sPos(1,N_s)^2-sPos(2,N_s)^2+estDist(N_s)^2));
end
solution = X\y;
ex = solution(1);
ey = solution(2);
```

Simulation outputs are;

- Emitter is found at (41.6799,65.686)
- Estimation Error is 2.9296e-14 meters

Placement of Sensors/Emitter and the estimated emitter location is shown in Figure 3.

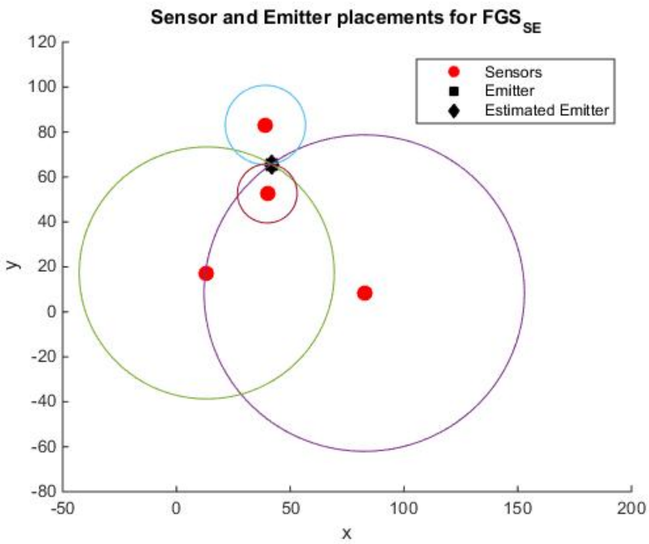

Figure 3: Placement of sensors and emitter in the region of interest and the estimation location using Maximum Likelihood with grid length 5, region of interest 100m x 100m, path loss exponent 3.5 and no additional noise. Estimation error is 2.9296e-14 meters, emitter is found at (41.6799,65.686).

2.3 Trilateration

Distances are estimated from each sensor. Circles are drawn such that center of the circles are the locations of the sensors and radius of the circles are the estimated distances. Emitter is estimated at the intersection of these circles. Mathematically this can be represented as solving the equation set shown in Equation 5 where x_k, y_k and r_k are k$^{\text{th}}$ sensors x coordinate, y coordinate and estimated distance from the emitter. Code for this implementation is a bit long therefore is not provided here.

$$\begin{cases} (x-x_1)^2+(y-y_1)^2=r_1^2 \\ \vdots \\ (x-x_k)^2+(y-y_k)^2=r_k^2 \end{cases} \tag{5}$$

Simulation outputs are;

- Emitter is found at (41.6799,65.686)
- Estimation Error is 2.1316e-14 meters

Placement of Sensors/Emitter and the estimated emitter location is shown in Figure 4.

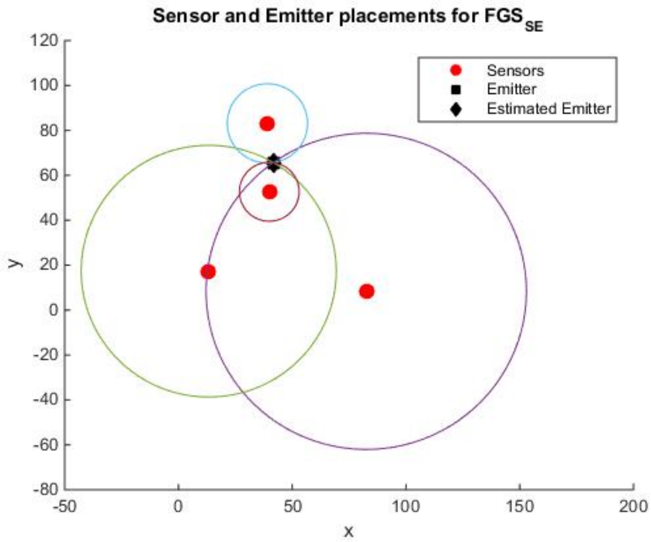

Figure 4: Placement of sensors and emitter in the region of interest and the estimation location using Triangulation with grid length 5, region of interest 100m x 100m, path loss exponent 3.5 and no additional noise. Estimation error is 2.9296e-14 meters, emitter is found at 2.1316e-14 meters.

3 Conclusion

RSS ranging techniques examined in [1] are implemented in Matlab and shown that they work under no noise conditions. It appears Min-Max method is capable of creating some large errors even when there is no additional noise. It might show some interesting results when tested in a noisy simulation so keeping it in the future simulations might prove fruitful. Maximum likelihood and Triangulation shows similar results under no noise conditions.

References

[1] E. Goldoni et al. "Experimental analysis of RSSI-based indoor localization with IEEE 802.15.4". In: *Wireless Conference (EW), 2010 European*. Apr. 2010, pp. 71–77. DOI: 10.1109/EW.2010.5483396.