

## INDOOR POSITIONING ALGORITHM

### BASED ON MLE-PSO WITH IMPROVED RSSI DISTANCE MODEL

#### 1) RSSI Distance Model:

Ta có:

$$RSSI_d = -10 * n * \log\left(\frac{d}{d_0}\right) + A + \xi \quad (1)$$

Trong đó  $RSSI_d$  là cường độ tín hiệu giữa máy thu và máy phát ở khoảng cách  $d$ .  $d_0$  là khoảng cách tham chiếu.  $A$  là cường độ tín hiệu ở khoảng cách tham chiếu.  $\xi$  là biến phân phối ngẫu nhiên Gaussian với trung bình 0 và phương sai  $\sigma^2$ .  $n$  là suy hao truyền dẫn.

Để thuận tiện trong tính toán, cho  $d_0 = 1m$  khi đó 1 trở thành:

$$RSSI_d = -10 * n * \log(d) + A + \xi \quad (2)$$

#### 2) RSSI Real-Time Correction Algorithm:

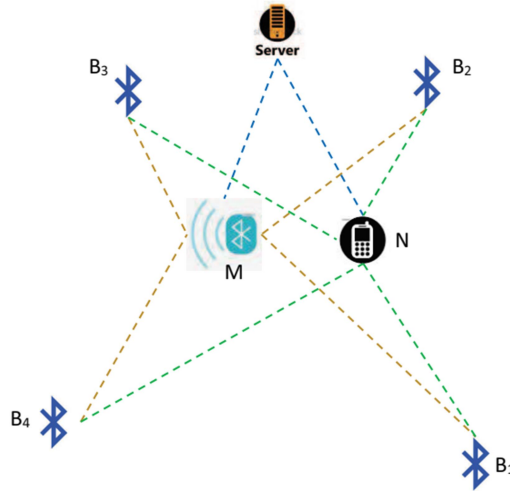


Fig1. Mô hình định vị trong nhà dùng thuật toán sửa sai RSSI

Trong thuật toán sửa sai RSSI theo thời gian thực, ngoài các **nút neo Bluetooth** và các **asset tag**, ta sử dụng thêm các **gateway Bluetooth** để sửa sai cho **asset tag**

Trong Fig1,  $B_1, B_2, B_3, B_4$  là các **nút neo Bluetooth**,  $N$  là **asset tag**,  $M$  là **gateway Bluetooth**.

Gọi  $\overline{R_{Ml}}$  là cường độ tín hiệu RSSI của gateway  $M$  đối với nút neo  $B_l$ . Ta có:

$$\overline{R_{Ml}} = -10 * n * \log(d_{Ml}) + \overline{A_l} \quad (3)$$

Trong đó  $d_{Ml}$  là khoảng cách từ  $M$  đến  $B_l$ .  $\overline{A_l}$  là trung bình cường độ tín hiệu ở khoảng cách 1m.

Vậy đối với hệ thống thời gian thực ta có:

$$R_{Ml} = -10 * n * \log(d_{Ml}) + A_l \quad (4)$$

Tương tự đối với tag N:

$$\overline{R_{Ml}} = -10 * n * \log(d_{Ml}) + \overline{A_l} \quad (5)$$

$$R_{Ml} = -10 * n * \log(d_{Ml}) + A_l \quad (6)$$

Từ (3) và (4) ta có:

$$R_{Ml} - \overline{R_{Ml}} = A_l - \overline{A_l} = \Delta A_l \quad (7)$$

Với  $\Delta A_l$  là biến động thời gian thực của hệ thống Bluetooth (bù hiệu chỉnh RSSI)

Tương tự với (5) và (6) ta có:

$$R_{Nl} - \overline{R_{Nl}} = A_l - \overline{A_l} = \Delta A_l \quad (8)$$

Vậy:

$$R_{Nl} = \Delta A_l + \overline{R_{Nl}} \quad (9)$$

Cuối cùng, RSSI  $\tilde{R}_{Nl}$  chính xác được cho bởi:

$$\tilde{R}_{Nl} = R_{Nl} - \Delta A_l = -10 * n * \log(d_{Nl}) - \Delta A_l \quad (10)$$

Bằng thuật toán đã trình bày ở trên, ta có thể loại bỏ 1 phần sai số do sự biến động năng lượng phát ra của Bluetooth và tính được RSSI Distance Model chính xác hơn.

### 3) Maximum Likelihood Estimation – Particle Swarm Optimization:

Từ phương trình (10) ở phần 2, ta có thể tính ra được  $d_{Nl}$ . Vậy bài toán bây giờ đặt ra là từ các  $d_{Nl}$  ( $l = 1, 2, 3, 4, \dots$ ) ta có thể tính được vị trí (x,y) của N nếu biết trước tọa độ của các  $B_l$ .

#### 3.1) Maximum Likelihood Estimation:

Giả sử ta có n beacons, vị trí của beacon thứ  $l$  là  $z_l = [x_l \ y_l]^T$ . Ngoài ra, cho  $z = [x \ y]^T$  là vị trí của asset tag.  $d_l$  là khoảng cách từ asset tag đến  $B_l$ . Như đã trình bày, ta có thể tìm được  $d_l$  theo thuật toán sửa sai RSSI. Vậy ta có:

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = d_1^2 \\ (x - x_2)^2 + (y - y_2)^2 = d_2^2 \\ \vdots \\ (x - x_n)^2 + (y - y_n)^2 = d_n^2 \end{cases} \quad (11)$$

Lấy các phương trình từ 1 đến n-1 lần lượt trừ cho phương trình thứ n ta được:

$$\begin{cases} x_1^2 - x_n^2 + y_1^2 - y_n^2 + 2(x_1 - x_n)x - 2(y_1 - y_n)y = d_1^2 - d_n^2 \\ x_2^2 - x_n^2 + y_2^2 - y_n^2 + 2(x_2 - x_n)x - 2(y_2 - y_n)y = d_2^2 - d_n^2 \\ \vdots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 + 2(x_{n-1} - x_n)x - 2(y_{n-1} - y_n)y = d_{n-1}^2 - d_n^2 \end{cases} \quad (12)$$

Viết lại hệ (12) theo  $AZ = b$ , ta có:

$$A = \begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) \\ 2(x_2 - x_n) & 2(y_2 - y_n) \\ \vdots & \vdots \\ 2(x_3 - x_n) & 2(y_3 - y_n) \end{bmatrix}$$

$$b = \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 + d_1^2 - d_n^2 \\ x_2^2 - x_n^2 + y_2^2 - y_n^2 + d_2^2 - d_n^2 \\ \vdots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 + d_{n-1}^2 - d_n^2 \end{bmatrix}$$

$$Z = \begin{bmatrix} x \\ y \end{bmatrix}$$

Cuối cùng ta có thể tính toán được kết quả của hệ phương trình:

$$\hat{Z} = (A^T A)^{-1} A^T b \quad (13)$$

### 3.2) Particle Swarm Optimization:

Particle Swarm Optimization (PSO) là một thuật toán tính toán tiến hóa thông minh được đề xuất bởi James Kennedy và Russell Eberhart vào năm 1995. PSO là một cách tiếp cận ngẫu nhiên dựa trên dân số mạnh mẽ để giải quyết các vấn đề tối ưu hóa toàn cục phi tuyến.

Một phần tử ngẫu nhiên sẽ được khởi tạo và lặp đi lặp lại để tìm kiếm giải pháp tối ưu nhất. Trong mỗi vòng lặp, các phần tử duy trì việc theo dõi vị trí tốt nhất của chính nó, được xem như **personal best (pbest)**, trong khi vị trí tốt nhất của toàn nhóm là **global best (gbest)**.

Mỗi phần tử sẽ thay đổi vị trí của nó dựa theo vận tốc hiện tại và khoảng cách giữa **pbest** và **gbest**. Theo cách này, dự kiến, **quần thể** sẽ tiến tới giải pháp tối ưu.

Quá trình lặp lại như sau:

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 [pbest_i(t) - z_i(t)] + c_2 r_2 [gbest(t) - z(t)] \quad (14)$$

$$Z_i(t+1) = Z_i(t) + V_i(t+1) \quad (15)$$

$$\omega = \omega_{max} - t \frac{\omega_{max} - \omega_{min}}{T} \quad (16)$$

Trong đó:  $V_i(t)$  là vận tốc của phần tử thứ  $i$ .  $Z_i(t)$  là vị trí của phần tử thứ  $i$ ,  $\omega$  là nhân tố quán tính.  $c_1, c_2$  là hệ số gia tốc,  $r_1, r_2$  là hệ số co thắt ngẫu nhiên trong phạm vi  $(0,1)$ .  $t$  là số thứ tự bước lặp hiện tại và  $T$  là tổng số bước lặp.

Bởi vì sai số là điều không thể tránh khỏi nên bản chất của việc xác định vị trí là hạn chế sai số, và việc ước tính vị trí có thể chuyển thành việc tìm được minimum fitness.

$$f_i(\hat{Z}) = d_i - |\hat{Z} - Z_i| \quad (17)$$

Chúng ta cập nhật quần thể bằng PSO, tối ưu hóa hàm fitness ở phương trình (18), kết thúc vòng lặp khi đạt được độ chính xác đề ra hoặc hoàn thành toàn bộ vòng lặp. Khi đó, phần tử tốt nhất chính là vị trí ước tính.

$$fitness\hat{Z} = \sum_{i=1}^n f_i^2(\hat{Z}) \quad (18)$$

### 3.3) System Model:

Bước 1: Hiệu chỉnh RSSI đo được để giảm thiểu nhiễu, sai số trong quá trình đo.

Bước 2: Tính khoảng cách giữa asset và beacon theo dữ liệu RSSI đã hiệu chỉnh ở bước 1.

Bước 3: Tính toán vị trí bằng MLE sử dụng khoảng cách đã tính ở bước 2, và giới hạn phạm vi ban đầu cho PSO.

Bước 4: Khởi tạo các thông số cần thiết: Đặt các hệ số gia tốc  $c_1, c_2$ , giá trị lớn nhất và nhỏ nhất của hệ số quán tính  $\omega_{max}, \omega_{min}$ . Số bước lặp  $T$  và số phần tử trong quần thể  $N$ .

Bước 5: Khởi tạo tập quần thể gồm  $N$  phần tử 1 cách ngẫu nhiên với giới hạn ở bước 3.

Bước 6: Đánh giá độ fitness của mỗi phần tử và cập nhật  $pbest_i(t)$  và  $gbest(t)$ .

Bước 7: Cập nhật vị trí và vận tốc của các phần tử trong quần thể theo các công thức (14) (15) (16)

Bước 8: Kiểm tra từng phần tử có nằm trong phạm vi giới hạn không, nếu không, đưa các phần tử này về lại trong phạm vi giới hạn:  $Z_i(t+1) = Z_i(t)$

Bước 9: Kiểm tra xem đã đạt đến độ chính xác yêu cầu hoặc hết vòng lặp hay chưa, nếu đã đạt đến thì kết thúc vòng lặp, xuất ra vị trí tối ưu, nếu không quay về bước 6.

Fig2 biểu diễn sơ đồ khối hệ thống của **MLE-PSO with the Improved RSSI Distance Model**.

