

INDOOR POSITIONING ALGORITHM BASED ON MLE-PSO WITH IMPROVED RSSI DISTANCE AND PATHLOSS MODEL

A) Giới thiệu đề tài:

Hệ thống định vị trong nhà (Indoor Positioning System – IPS) là một mạng lưới các thiết bị được sử dụng để định vị người hoặc vật thể trong đó GPS và các công nghệ vệ tinh khác thiếu độ chính xác hoặc thất bại hoàn toàn (như bên trong các tòa nhà nhiều tầng, sân bay, tầng hầm). Rất nhiều kỹ thuật và thiết bị được sử dụng để định vị trong nhà, ví dụ như smartphone, WIFI và Bluetooth, camera.

Các công nghệ IPS đòi hỏi cơ sở hạ tầng phải là các tần số vô tuyến như là RFID hoặc UWB (Ultra Wide Band), hồng ngoại, siêu âm, Zigbee, laser. Ngoài ra IPS còn có thể sử dụng các cơ sở hạ tầng khác của tòa nhà như là WIFI, sóng di động, Bluetooth.

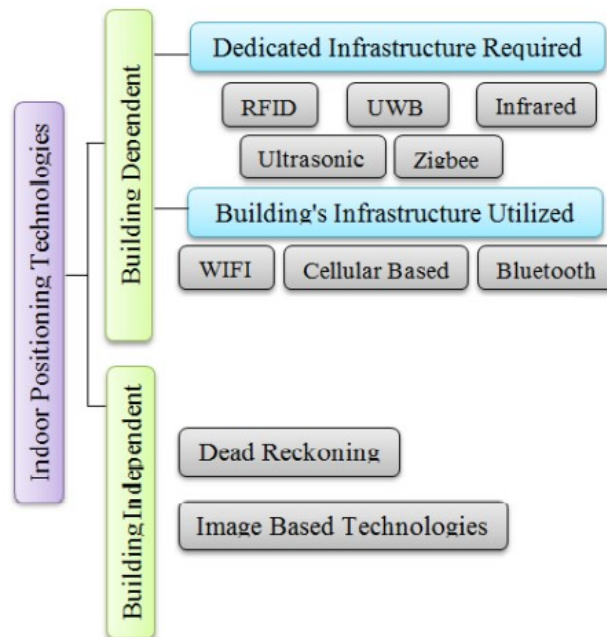


Fig0. Phân loại các công nghệ định vị trong nhà

Ở bài báo cáo này sẽ sử dụng công nghệ Bluetooth và thuật toán tính khoảng cách từ Receive Signal String Indicator (RSSI) từ đó xác định vị trí vật thể.

Bluetooth là một chuẩn công nghệ truyền thông không dây tầm gần giữa các thiết bị điện tử. Công nghệ này hỗ trợ việc truyền dữ liệu qua các khoảng cách ngắn giữa các thiết bị di động và cố định, tạo nên các mạng cá nhân không dây (Wireless Personal Area Network-PANs).

Bluetooth có thể đạt được tốc độ truyền dữ liệu 1Mb/s. Bluetooth hỗ trợ tốc độ truyền tải dữ liệu lên tới 720 Kbps trong phạm vi 10 m–100 m. Khác với kết nối hồng ngoại (IrDA), kết nối Bluetooth là vô hướng và sử dụng giải tần 2,4 GHz.

Bluetooth Low Energy (BLE), hay còn được biết đến là “Bluetooth Smart”, công nghệ này được giới thiệu khi BLE 4.0 chính thức được ra mắt. Như vậy từ BLE 4.0 bên cạnh Bluetooth classic, thì chúng ta đã có một người anh em khác là BLE Smart. Hai công nghệ này có những điểm ưu và hạn chế khác nhau, tùy vào yêu cầu kỹ thuật mà ta lựa chọn công nghệ cho phù hợp. Bluetooth Low Energy còn được gọi là Bluetooth năng lượng thấp, Bluetooth công suất thấp, là công nghệ mạng cá nhân. Hiện nay, có rất nhiều công nghệ không dây được giới thiệu, tuy nhiên BLE vẫn được yêu thích sử dụng trong nhiều dự án, với lý do nó có tính tương thích cao với nhiều nền tảng. Đặc biệt với các nền tảng mobile như (iOS, Android, Windows Phone,...).

BLE (Bluetooth 4.0 trở đi) được thiết kế cho các ứng dụng:

- Siêu tiết kiệm năng lượng, cho phép thiết bị hoạt động trong vài tháng hoặc vài năm chỉ với một viên pin đồng xu (coin-cell battery);
- Khoảng cách ngắn, hoạt động ổn định trong phạm vi 10m
- Dữ liệu truyền tải không lớn, thích hợp cho các ứng dụng điều khiển không liên tục, cảm biến.
- Các ứng dụng điển hình sử dụng BLE như thiết bị theo dõi sức khỏe, beacons, nhà thông minh, an ninh, giải trí, cảm biến tiệm cận, ô tô. Trung tâm của một hệ thống ứng dụng BLE thường là Smart phones, tablets và PCs

Beacon là thiết bị hoạt động trên công nghệ BLE.. Ứng dụng tiềm năng là một địa điểm nhận thức, nhận biết ngữ cảnh, cảm biến nhỏ không dây có thể xác định vị trí của người sử dụng trong một cửa hàng.

Beacon có thể gửi thông báo của các mặt hàng gần đó được bán hoặc những mặt hàng khách hàng có thể tìm kiếm, và nó có thể cho phép thanh toán tại các điểm bán hàng (POS) mà khách hàng không cần phải mở ví ra hoặc đưa thẻ của họ để thực hiện thanh toán. Nó là một đối thủ cạnh tranh của Near Field Communication (Công nghệ NFC).

Từ những thông tin trên, ta có thể hình thành một đề tài định vị trong nhà như sau:

- Ta sẽ sử dụng Beacon là như trạm phát các tín hiệu bluetooth gửi đến các trạm thu khác

- Các trạm thu (ví dụ như ESP32 hỗ trợ WIFI và BLUETOOTH) sẽ nhận các tín hiệu, đồng thời đo RSSI và gửi các thông số này lên server.
- Trên server sẽ tổng hợp các dữ liệu, sử dụng thuật toán đã được cài đặt sẵn tính ra được vị trí của vật thể.

Ở đây, ta có thể sử dụng Beacon gắn liền với vật cần xác định và các trạm thu được đặt ở vị trí cố định (reference points) hoặc ngược lại. Trong thuật toán được trình bày dưới đây, ta sẽ sử dụng mô hình Beacons – References Points, ESP32 – Asset Tag và Gateway.

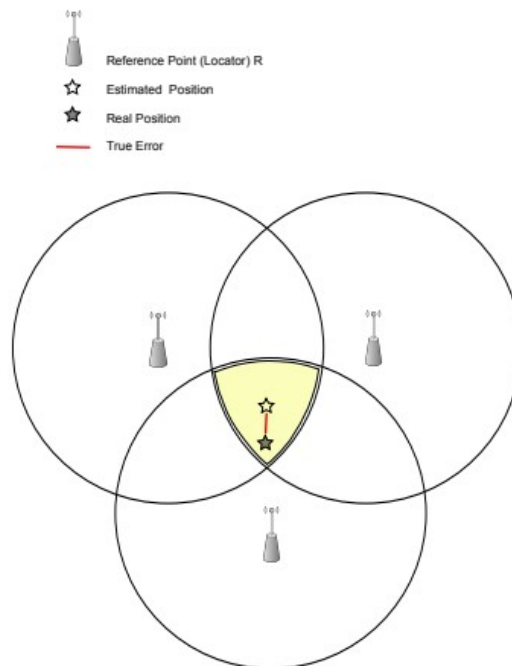


Fig1. Định vị bằng các điểm tham chiếu

B) MLE-PSO With Improved RSSI Distance and Pathloss 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. ξ là biến phân phối ngẫu nhiên Gaussian với trung bình 0 và phương sai σ^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:

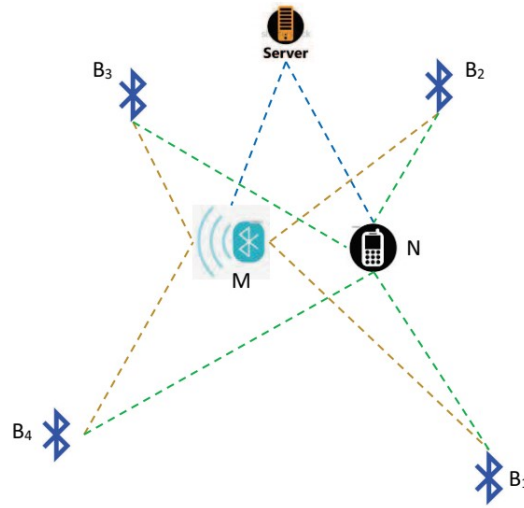


Fig2. Mô hình định vị trong nhà dùng thuật toán sửa sai RSSI

Trong thuật toán sửa sai RSSI theo thời gian thực, ngoài các **nút neo Bluetooth** (Beacons Reference Points) và các **asset tag** (ESP32), ta sử dụng thêm các **gateway Bluetooth** (ESP32) 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 ΔA_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:

$$\begin{cases} x_1^2 - x_n^2 + y_1^2 - y_n^2 - 2x_1x_n + 2x_n^2 - 2y_1y_n + 2y_n^2 = d_1^2 - d_n^2 \\ x_2^2 - x_n^2 + y_2^2 - y_n^2 - 2x_2x_n + 2x_n^2 - 2y_2y_n + 2y_n^2 = d_2^2 - d_n^2 \\ \vdots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 - 2x_{n-1}x_n + 2x_n^2 - 2y_{n-1}y_n + 2y_n^2 = d_{n-1}^2 - d_n^2 \end{cases} \quad (12)$$

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 , ω 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) Kalman Filter:

Bộ lọc Kalman, được Rudolf (Rudy) E. Kálmán công bố năm 1960, là thuật toán sử dụng chuỗi các giá trị đo lường, bị ảnh hưởng bởi nhiễu hoặc sai số, để ước đoán biến số nhằm tăng độ chính xác so với việc sử dụng duy nhất một giá trị đo lường. Bộ lọc Kalman thực hiện phương pháp truy hồi đối với chuỗi các giá trị đầu vào bị nhiễu, nhằm tối ưu hóa giá trị ước đoán trạng thái của hệ thống.

Bộ lọc Kalman được ứng dụng rộng rãi trong kỹ thuật, phổ biến trong các ứng dụng định hướng, định vị và điều khiển các phương tiện di chuyển. Ngoài ra, bộ lọc Kalman còn được ứng dụng để phân tích dữ liệu trong các lĩnh vực xử lý tín hiệu và kinh tế.

Trong mô hình IPS này, chúng ta sẽ sử dụng Kalman Filter vào dữ liệu RSSI đo được để hạn chế sai số và nhiễu ảnh hưởng.

Ta có mô hình cơ bản như sau:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (19)$$

Trong đó:

x_k là vector trạng thái (trong trường hợp này là RSSI) tại điểm k .

A là ma trận mô hình chuyển đổi trạng thái.

B là ma trận mô hình điều khiển đầu vào.

u_{k-1} là vector điều khiển tại thời điểm $k - 1$.

w_{k-1} vector ngẫu nhiên nhiễu hệ thống tại thời điểm $k - 1$.

Cùng với vector đo lường, ta có đủ phương trình cần thiết cho Kalman Filter:

$$z_k = Hx_k + v_k \quad (20)$$

Trong đó:

z_k là vector đo lường (RSSI đo được) tại thời điểm k .

H là ma trận mô hình quan sát.

v_k là vector ngẫu nhiên nhiễu đo lường.

Vì chúng ta đang sử dụng hệ thống tĩnh, không có sự tác động vào đối tượng nên vector điều khiển sẽ bằng 0 ($u_k = 0$). Ở đây, chúng ta áp dụng Kalman Filter vào RSSI thô đo được, nên vector trạng thái và vector đo lường sẽ chỉ có 1 phần tử, A và H sẽ bằng 1. Vậy ta có thể viết phương trình (19) (20) lại như sau:

$$x_k = x_{k-1} + w_{k-1} \quad (21)$$

$$z_k = x_k + v_k \quad (22)$$

Ngoài ra, ta có hàm phân phối nhiễu hệ thống và đo lường như sau:

$$p(w) \sim N(0, Q) \quad (21)$$

$$p(v) \sim N(0, R) \quad (22)$$

Trong đó:

N là hàm phân phối Gaussion.

Q là hiệp phương sai (covariance) nhiễu hệ thống.

R là hiệp phương sai nhiễu đo lường

Ở đây, chúng ta không đi sâu vào cách hình thành từng bước Kalman Filter, nên ta sẽ đi luôn vào cách sử dụng Kalman Filter như hình dưới.

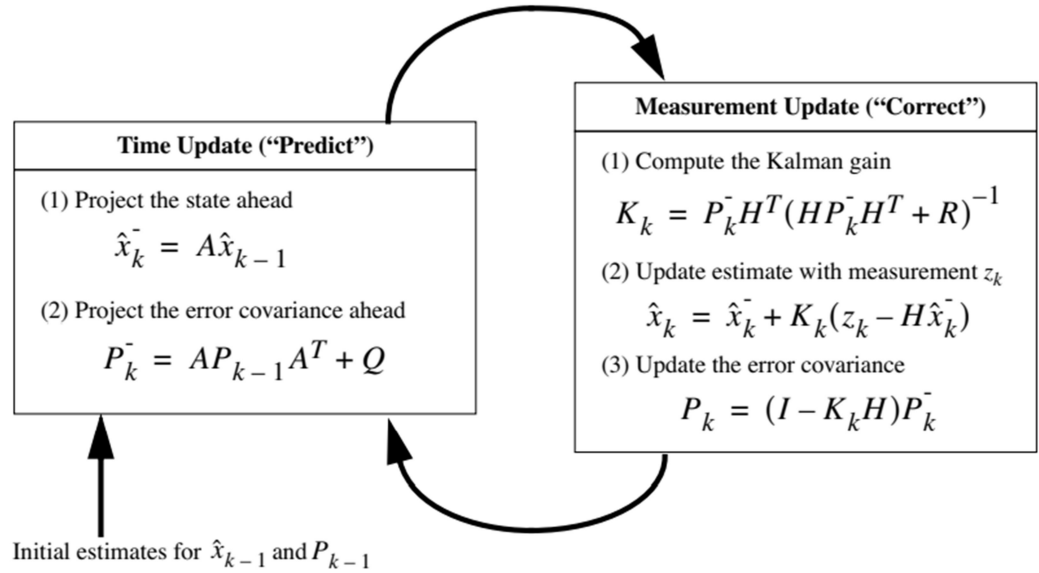


Fig3. Mô hình hoạt động của Kalman Filter

Trong đó:

P_k là hiệp phương sai ước lượng (estimate covariance).

K_k là độ lợi Kalman (Kalman Gain)

Theo Fig3, chúng ta cần phải khởi tạo x_{k-1} và P_{k-1} . Vậy ta phải cung cấp 5 tham số x_{k-1} , P_{k-1} , K_{k-1} , Q và R để mô hình Kalman Filter hoạt động:

x_{k-1} ta khởi tạo bằng với dữ liệu RSSI đo được lần đầu tiên.

P_{k-1} ta khởi tạo bằng với Q .

Q không có cách cài đặt cụ thể. Thông qua thử nghiệm để lấy Q tốt nhất.

R ta lấy được bằng cách lấy dữ liệu thô và đánh giá sai số $\pm \Delta RSSI$.

3.4) System Model:

Bước 1: Hiệu chỉnh RSSI đo được để giảm thiểu nhiễu, sai số trong quá trình đo, sau đó đưa vào bộ lọc Kalman.

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.

Fig4 biểu diễn sơ đồ khối hệ thống của **MLE-PSO with the Improved RSSI Distance Model**.

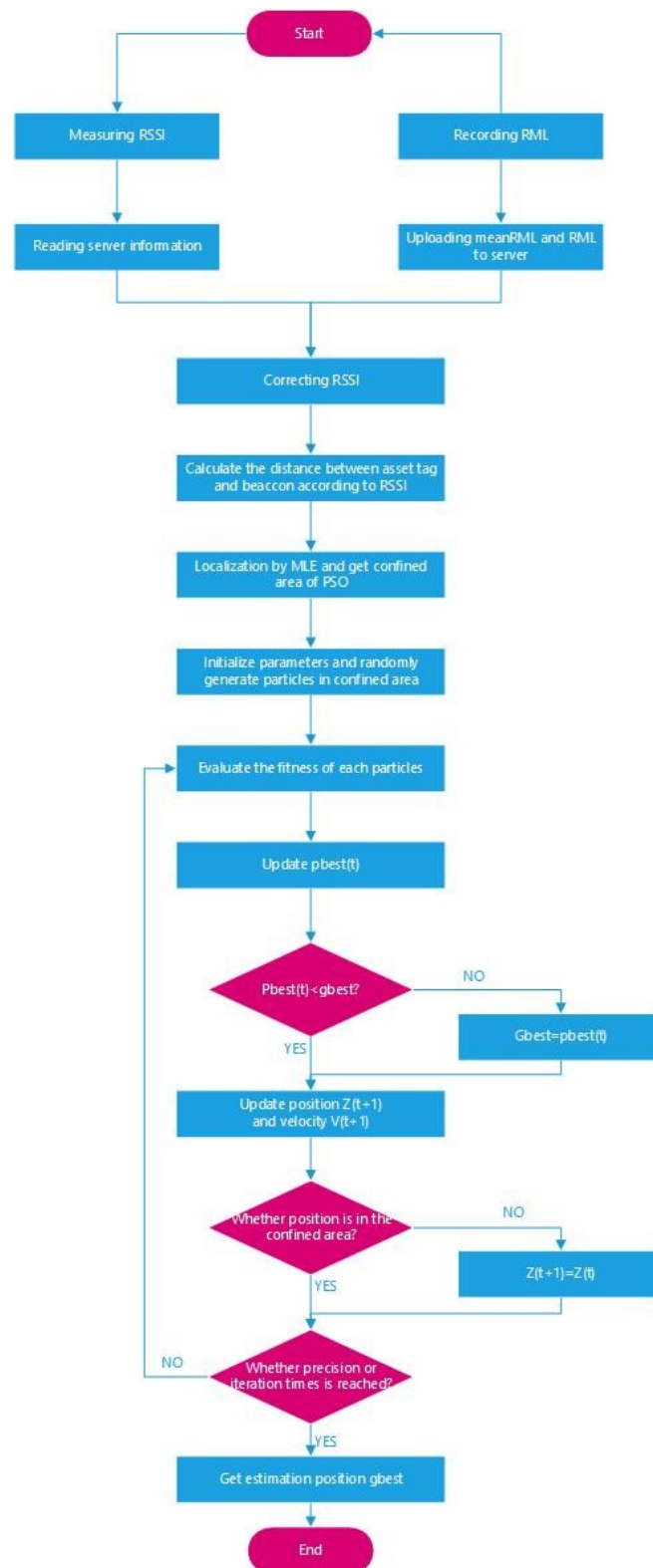


Fig4. System Model

4) Improving Pathloss Algorithm:

Như ta đã biết ở công thức (1), mô hình tính khoảng cách dự trên RSSI có phụ thuộc vào một tham số môi trường là **Suy hao truyền dẫn (Pathloss)**.

Path Loss, hay **Path Attenuation**, là sự giảm mật độ năng lượng (suy giảm) của sóng điện từ khi nó truyền qua không gian. Path Loss là một thành phần chính trong phân tích và thiết kế ngân sách liên kết của một hệ thống viễn thông.

Thuật ngữ này thường được sử dụng trong truyền thông không dây và truyền tín hiệu. Path Loss có thể do nhiều tác động, chẳng hạn như mất không gian tự do, khúc xạ, nhiễu xạ, phản xạ, mất khớp nối khẩu độ-trung bình và hấp thụ. Path Loss cũng bị ảnh hưởng bởi các đường viền địa hình, môi trường (thành thị hoặc nông thôn, thảm thực vật và tán lá), môi trường lan truyền (không khí khô hoặc ẩm), khoảng cách giữa máy phát và máy thu, và chiều cao và vị trí của ăng ten.

Ở các môi trường khác nhau, ta có thành phần Path Loss là khác nhau. Bảng 1 cho ví dụ về các môi trường điển hình:

| Environment | Path Loss Exponent, n |
|-------------------------------|-----------------------|
| Free space | 2 |
| Urban area cellular radio | 2.7 to 3.5 |
| Shadowed urban cellular radio | 3 to 5 |
| In building line-of-sight | 1.6 to 1.8 |
| Obstructed in building | 4 to 6 |
| Obstructed in factories | 2 to 3 |

Table1. Pathloss Exponent ở các môi trường khác nhau.

Vậy vấn đề cần đặt ra là: Nếu muốn độ chính xác thuật toán càng cao, thì yếu tố phụ thuộc môi trường phải giảm thiểu. Yếu tố môi trường ở thuật toán này chính là **Pathloss Exponent**.

Ta nhận thấy theo công thức (1), nếu ta biết được khoảng cách giữa trạm thu và trạm phát, RSSI thu được, RSSI tại 1m, ta sẽ tính ngược lại được tham số Pathloss Exponent.

Vậy để thuận tiện, ta thêm 1 Refence Point với vai trò là trạm thu đã biết trước vị trí. Sử dụng thuật toán **RSSI Real-Time Correction** đã trình bày ở trên thì ta sẽ có được các thông số cần tìm từ đó tìm được thành phần Pathloss Exponent và thay vào phương trình (10) và tìm ra được khoảng cách có độ chính xác cao hơn.

C) Testing Model:

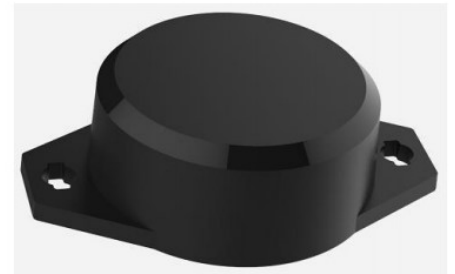
Để thử nghiệm cho mô hình ở trên, như đã trình bày, ta sử dụng các Beacon làm trạm phát (Reference Points), Asset Tag là trạm thu (Moving), Gateway và Pathloss Gateway là trạm thu (Reference Points).

Các thiết bị được lựa chọn để sử dụng:

Beacon ta sử dụng DearBeacon #9 của Minew (<https://www.minew.com/>)

FEATURE

- IP67 waterproof;
- Replaceable coin battery 1000mAh
- Advertising iBeacon & Eddystone & Sensordata
- Bluetooth® 5.0 chipset nRF52 series
- The max. 100 meters advertising distance
- Internal ON/OFF push button
- Tough enclosure and screw fixing
- Temperature & accelerometer sensor option



DearBeacon E9

Asset Tag, Gateway và Pathloss Gateway ta sử dụng ESP32 của Espressif (<https://espressif.com/>):



Ở Đề cương luận văn tốt nghiệp, ta sẽ dừng ở phần tính khoảng cách giữa trạm thu và trạm phát. Nếu khoảng cách có độ chính xác càng cao thì khi đưa vào MLE-PSO sẽ có kết quả vị trí cần tìm càng chính xác.

Bố trí các thiết bị ở vị trí phù hợp, tiến hành đo khoảng cách vật lý và thử nghiệm mô hình, ta có các kết quả sau:



Fig4. Thông số RSSI Mean của Gateway tích lũy được theo thời gian.



Fig5. Thông số RSSI của Asset Tag đo được



Fig6. Thông số RSSI của Asset Tag sau khi được Correct



Fig7. Thông số RSSI của Asset tag sau khi được Correct và đưa vào bộ lọc Kalman

Đặt Asset Tag ở khoảng cách 4.8m, Pathloss Gateway ở khoảng cách 4.2 m. Ta được kết quả như sau: (Ở giai đoạn đầu và giữa là thời gian để RSSI Mean của Gateway xác lập, ở phần sau, sau khi đã xác lập thì khoảng cách đã được tính chính xác ở với sai số không lớn)

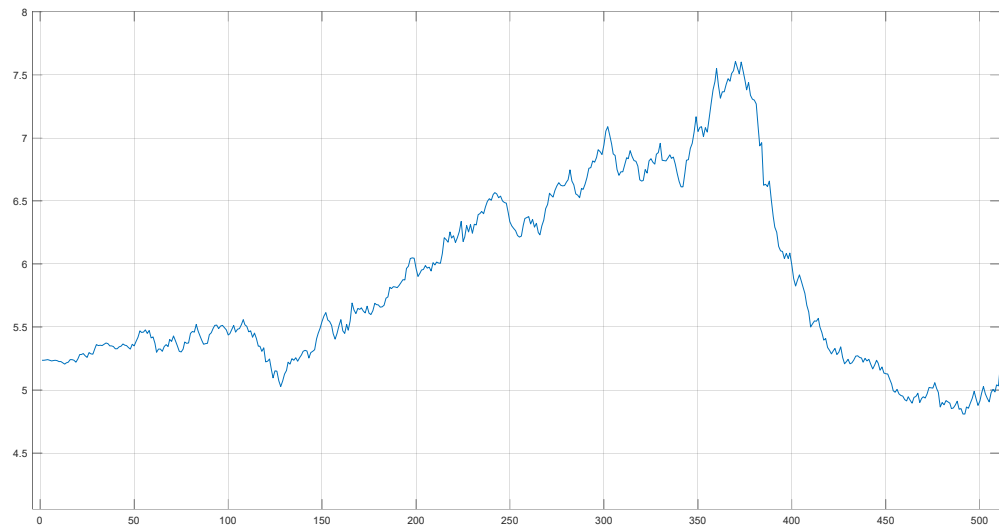


Fig8. Khoảng cách tính được khi Asset cách 4.8m

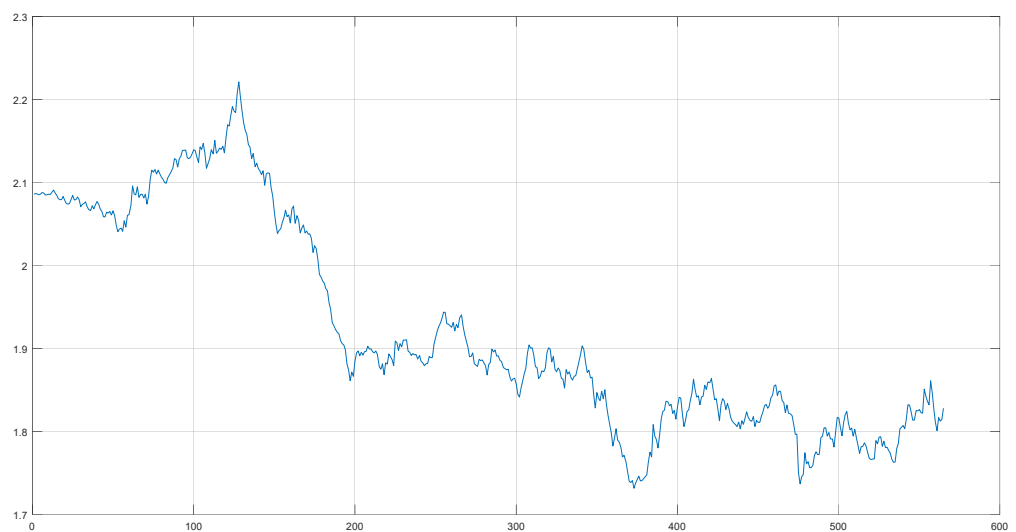


Fig9. Pathloss tính được khi Asset Tag cách 4.8m và Pathloss Gateway cách 4.2m

Tiếp tục thử lại với khoảng cách [4.8m, 4.2m]:

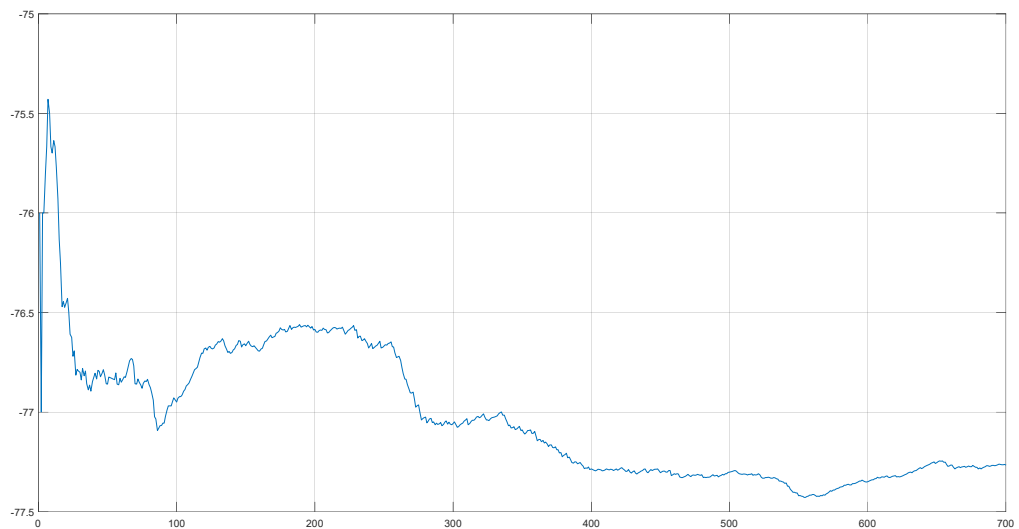


Fig10. RSSI Mean [4.8, 4.2]

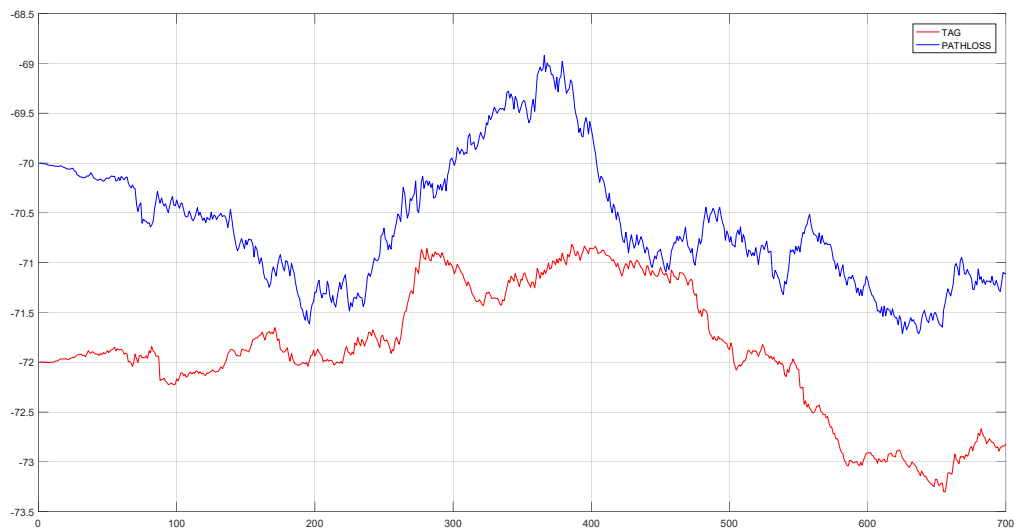


Fig11. RSSI Asset Tag và Pathloss Gateway sau khi đã Correct và đưa vào Kalman ở [4.8, 4.2]

Ta có thể nhận thấy ở phần cuối, tính chất của RSSI Correct và Kalman của cả 2 phần Tag và Pathloss đều như nhau. Từ đây có thể thấy thuật toán đang đi đúng hướng.

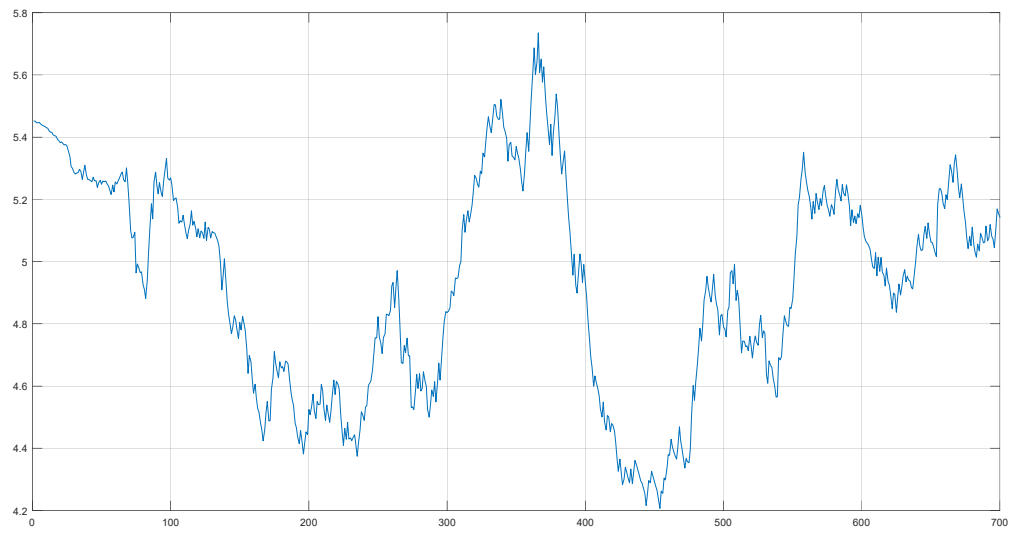


Fig12. Distance [4.8, 4.2]

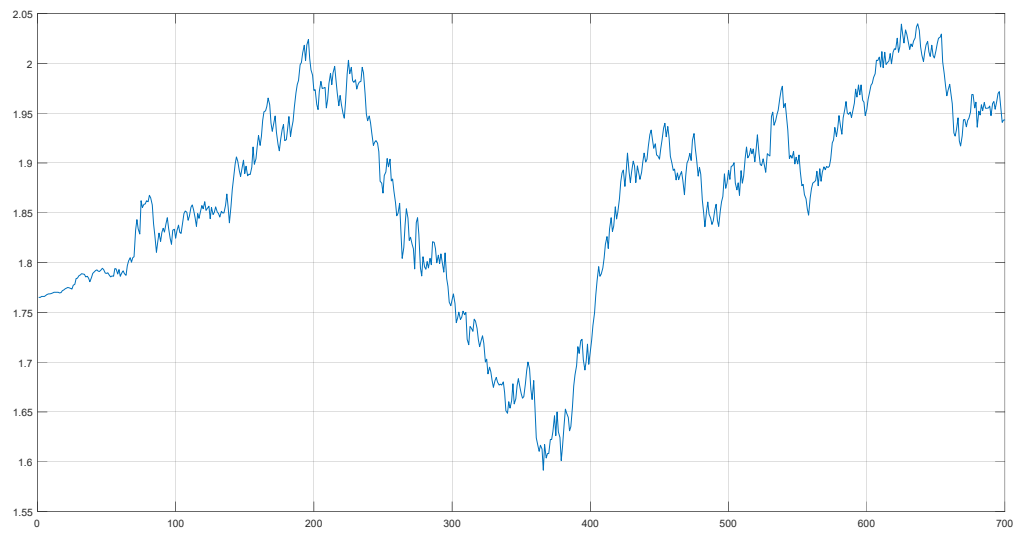


Fig13. Pathloss Exponent [4.8, 4.2]

Tiếp tục ta đặt ở khoảng cách [7.8m, 9m] ta được kết quả sau. (Chưa hoàn thiện)

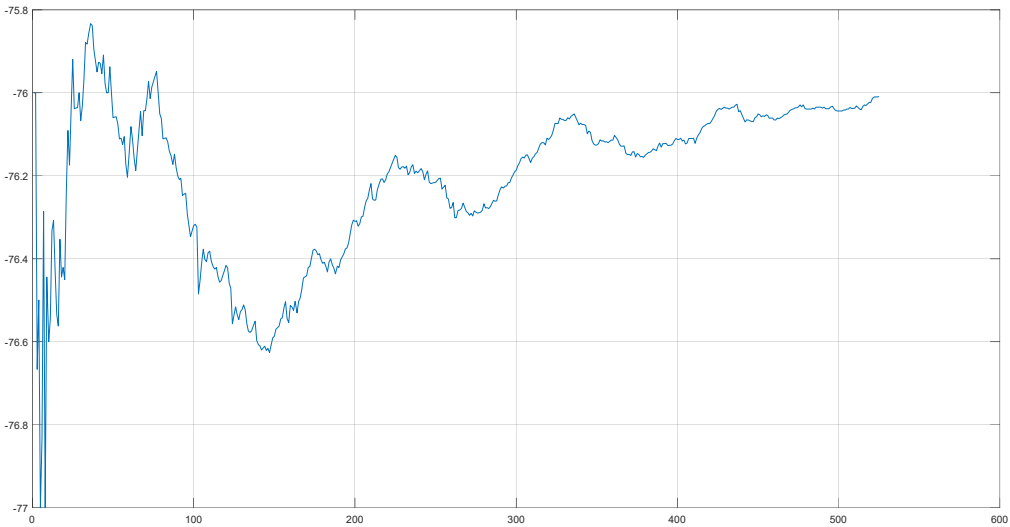


Fig14. RSSI Mean [9, 7.8]

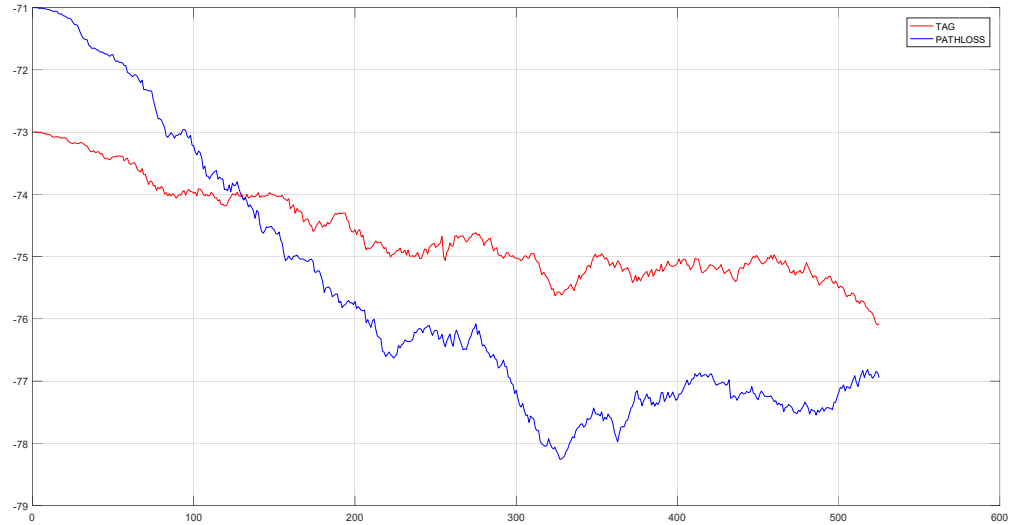


Fig15. RSSI Correct Kalman [9, 7.8]

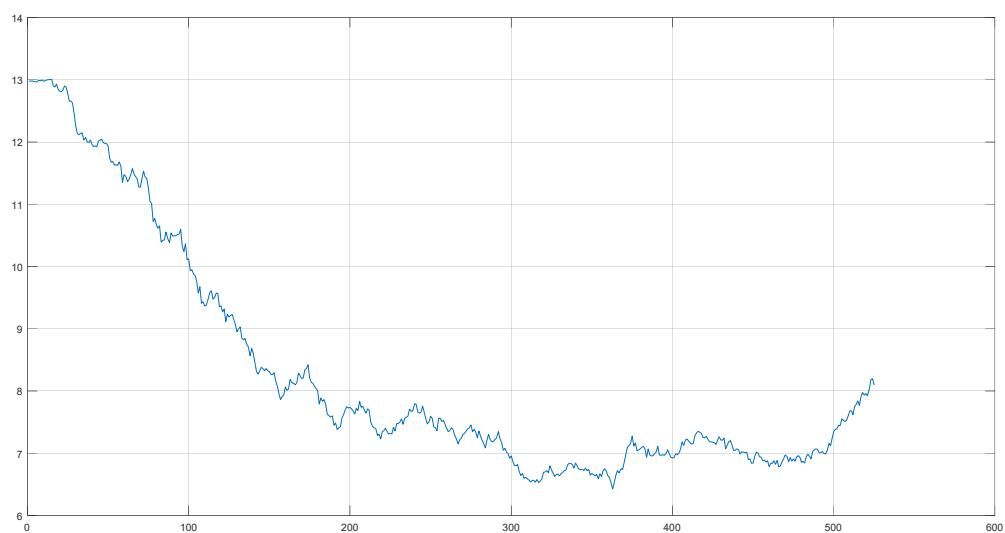


Fig16. Distance [9, 7.8]

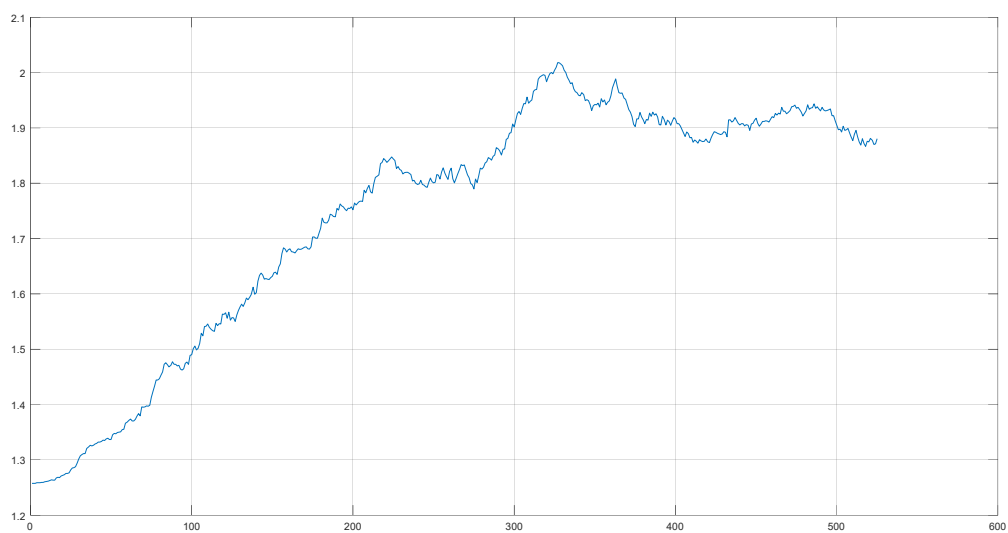


Fig17. Pathloss Exponent [9, 7.8m]

D) Nhận xét, Hướng phát triển Luận văn và Tài liệu tham khảo:

1) Nhận xét:

Model thử nghiệm chưa chạy được nhiều nên dữ liệu có phần thiếu sót, chưa làm đến cùng.

Trình bày thuật toán chưa thực sự rõ ràng. Cần nêu thêm chi tiết về giải thuật.

Bước đầu thử nghiệm cho kết quả khá khả quan.

2) Hướng phát triển Luận văn:

Cần phải tiếp tục thử nghiệm nhiều hơn, thử nghiệm ở các môi trường khác nhau để đưa ra kết quả đánh giá chính xác nhất. Từ đó chỉnh sửa giải thuật.

Sau khi có khoảng cách, cần xây dựng code cho MLE-PSO để đưa ra kết quả cần tìm cuối cùng, từ đó mới có thể đánh giá được tổng thể thuật toán và đưa ra giải pháp phù hợp.

3) Tài liệu tham khảo:

- DearBeacon, Model: E9, Datasheet V2, MINEW (<https://www.minew.com/>)
- ESP32, Espressif (<https://www.espressif.com/>)
- *A MLE-PSO Indoor Localization Algorithm Based On RSSI*, Chong Zhao1, Bo Wang1.
- *Chaotic Particle Swarm Optimization*, Yanxia Sun, Guoyuan Qi, Zenghui Wang, Barend Jacobus van Wyk, Yskandar Hamam.
- *A chaos particle swarm optimization ranging correction location in complex environment*, REN Wei-zheng, CAO Jing.
- *Indoor Positioning Algorithm Based on the Improved RSSI Distance Model*, Guoquan Li, Enxu Geng, Zhouyang Ye, Yongjun Xu, Jinzhao Lin and Yu Pang