Chương 4. Áp dụng các giải thuật tối thiểu sai số

Mục đích của chúng ta là tìm vị trí gần với vị trí nhất để có thể theo dõi, quản lý, vì vậy thực hiện phép toán phương trình (3.10) là chưa đủ bởi vì hệ phương trình (3.8) chỉ là phương trình suy ra từ hệ phương trình (3.4), có nghĩa là đáp số của hệ (3.8) chưa chắc là kết quả tối ưu nhất của hệ (3.4) vì vậy ta phải sử dụng thêm các giải thuật khác để tìm được kết quả tối ưu.

Trong chương này ta sẽ tìm hiểu và áp dụng các thuật toán để tối ưu sai số như bộ lọc **Kalman**, phương pháp **Pathloss Exponent Improvement** và giải thuật **Particle Swarm Optimization**.

4.1 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.

Ứng dụng đầu tiên và nổi tiếng nhất chính là bộ lọc Kalman đã được áp dụng để điều hướng cho Dự án Apollo, trong đó yêu cầu ước tính quỹ đạo của tàu vũ trụ có người lái lên Mặt trăng và quay trở lại Trái đất.

Mặc dù Bộ lọc Kalman được ứng dụng trong nhiều lĩnh vực, nhưng nó được sử dụng chủ yếu với 2 mục đích chính:

- **Estimating the state of dynamic system** (Ước tính trạng thái của hệ thống động) – trong đó, hệ thống động là hệ thống có trạng thái thay đổi theo thời gian, mà trong vũ trụ này thì hiếm có thứ nào hoàn toàn bất biến. Từ những thông tin chứa đầy nhiễu và sự không chắc chắn (noise & uncertainty), bộ lọc Kalman có thể cung cấp cho chúng ta các giá trị ước tính (chính xác nhất có thể) về trạng thái hiện tại của hệ thống.
- **The Analysis of Estimation Systems** (Phân tích hệ thống dự đoán) Với các thông tin, giá trị ước tính về trạng thái hiện tại của hệ thống, bộ lọc kalman còn có thể tiên đoán được các giá trị ở trạng thái tiếp theo của hệ thống.

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:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (4.1)$$

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:

$$z_k = Hx_k + v_k \quad (4.2)$$

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 lại phương trình (4.1) và (4.2) như sau:

$$x_k = x_{k-1} + w_{k-1} \tag{4.3}$$

$$z_k = x_k + v_k \tag{4.4}$$

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

$$\begin{aligned} p(w) &\sim N(0, Q) \\ p(v) &\sim N(0, R) \end{aligned} \tag{4.5}$$

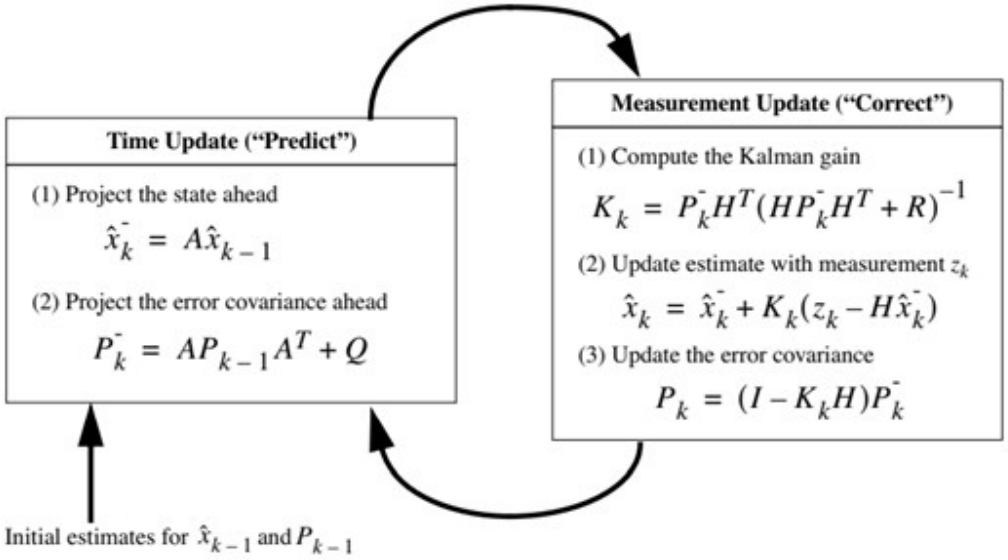
Trong đó:

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.



Hình 4.1: 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 như hình 4.1, chúng ta cần 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 bộ lọc Kalman hoạt động.

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

K_{k-1}ta sẽ cho bằng 0, sẽ tự động thay đổi sau các bước tính.

P_{k-1}ta khởi tạo bằng với Q, sẽ tự động thay đổi sau các bước tính.

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 RSSI thô và đánh giá sai số $\pm\Delta RSSI$.

4.2 Pathloss Exponent Improvement

Như ta đã biết trong chương 3, **Pathloss Exponent** là một tham số quan trọng để xây dựng phương trình RSSI Distance.

Path Loss, hay Path Attenuation, là sự giảm mật độ 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. năng lượng (suy giảm) của sóng điện từ khi nó truyền qua không gian.

Ở các môi trường khác nhau, ta có thành phần Path Loss là khác nhau. Bảng 4.1và 4.2 cho ví dụ về các môi trường điển hình:

Environment	Freq (MHz)	n
Indoor – Retail Store	914	2.2
Indoor – Grocery Store	914	1.8
Indoor – Hard Partition Office	1500	3.0
Indoor – Soft Partition Office	900	2.4
Indoor – Factory (LOS(*))	1900	2.6
Indoor – Factory (LOS)	1300	1.6-2.0
Indoor – Suburban Home	4000	2.1
Indoor – Factory (Obstructed)	1300	3.3
Indoor – Factory (Obstructed)	4000	2.1
Indoor – Office Same Floor	914	2.68-4.01
Indoor – Office Entire Building	914	3.54-4.33
Indoor – Office Wing	914	2.68-4.01

Bảng 4.1: Pathloss Exponent cho các môi trường cụ thể

Environment	Freq (MHz)	n
Indoor – Average	914	3.14
Indoor – Through One Floor	914	4.19
Indoor – Through Two Floor	914	5.04
Indoor – Through Three Floor	914	5.22

(*)LOS: Line of Sight

Bảng 4.2: Pathloss Exponent cho các môi trường cụ thể (tiếp theo)

Qua bảng trên ta có thể thấy, tham số Pathloss phụ thuộc khá nhiều vào môi trường xung quanh, ngoài ra còn có tần số của tín hiệu. Vậy nếu chúng ta đặt Pathloss Exponent là một hằng số cố định thì có thể sẽ gây ra sai số khá lớn, từ đó làm mất ổn định hệ thống, khó có thể phán đoán chính xác được vị trí thiết bị.

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.

Để giải quyết vấn đề trên, ta sẽ đặt thêm một thiết bị thứ hai. Thiết bị này sẽ được cố định tại một vị trí cụ thể, từ đó có thể suy ra được tham số Pathloss của môi trường.

Biến đổi phương trình (3.2) ta được:

$$n = \frac{A - RSSI_d}{10 * \log d} \tag{4.6}$$

Vậy với A (RSSI tại 1m) và d đã được cố định sẵn thì ta chỉ cần đo được $RSSI_d$ và áp dụng công thức (4.6) ta sẽ có được Pathloss Exponent và đưa tham số này vào phương trình tính khoảng cách ta sẽ có kết quả chính xác hơn từ đó tối ưu được vị trí cần tính.

4.3 Particle Swarm Optimization

4.3.1 Giới thiệu về giải thuật di truyền và tối ưu hóa bầy đàn

Giải thuật di truyền (Genetic Algorithm – GA) là kỹ thuật phỏng theo quá trình thích nghi tiến hóa của các quần thể sinh học dựa trên học thuyết Darwin. GA là phương pháp tìm kiếm tối ưu ngẫu nhiên bằng cách mô phỏng theo sự tiến hóa của con người hay của sinh vật. Tư tưởng của thuật toán di truyền là mô phỏng các hiện tượng tự nhiên, là kế thừa và đấu tranh sinh tồn.

GA thuộc lớp các giải thuật xuất sắc nhưng lại rất khác các giải thuật ngẫu nhiên vì chúng kết hợp các phần tử tìm kiếm trực tiếp và ngẫu nhiên. Khác biệt quan trọng giữa tìm kiếm của GA và các phương pháp tìm kiếm khác là GA duy trì và xử lý một tập các lời giải, gọi là một quần thể (population).

Trong GA, việc tìm kiếm giả thuyết thích hợp được bắt đầu với một quần thể, hay một tập hợp có chọn lọc ban đầu của các giả thuyết. Các cá thể của quần thể hiện tại khởi nguồn cho quần thể thế hệ kế tiếp bằng các hoạt động lai ghép và đột biến ngẫu nhiên – được lấy mẫu sau các quá trình tiến hóa sinh học.

Ở mỗi bước, các giả thuyết trong quần thể hiện tại được ước lượng liên hệ với đại lượng thích nghi, với các giả thuyết phù hợp nhất được chọn theo xác suất là các hạt giống cho việc sản sinh thế hệ kế tiếp, gọi là cá thể (individual). Cá thể nào phát triển hơn, thích ứng hơn với môi trường sẽ tồn tại và ngược lại sẽ bị đào thải. GA có thể dò tìm thế hệ mới có độ thích nghi tốt hơn.

GA giải quyết các bài toán quy hoạch toán học thông qua các quá trình cơ bản: lai tạo (crossover), đột biến (mutation) và chọn lọc (selection) cho các cá thể trong quần thể. Dùng GA đòi hỏi phải xác định được: khởi tạo quần thể ban đầu, hàm đánh giá các lời giải theo mức độ thích nghi – hàm mục tiêu, các toán tử di truyền tạo hàm sinh sản.

Phương pháp tối ưu bầy đàn (Particale Swarm Optimization – PSO) là một dạng của thuật toán tiến hóa quần thể như giải thuật di truyền (GA). Tuy vậy PSO khác với GA ở chỗ nó thiên về sử dụng sự tương tác giữa các cá thể trong một quần thể để khám phá không gian tìm kiếm. PSO là kết quả của sự mô hình hóa việc đàn chim bay đi tìm kiếm thức ăn cho nên nó thường được xếp vào các loại thuật toán có sử dụng trí tuệ bầy đàn.

Được giới thiệu vào năm 1995 tại một hội nghị của IEEE bởi James Kennedy và kỹ sư Russell C. Eberhart. Thuật toán có nhiều ứng dụng quan trọng trong tất cả các lĩnh vực mà ở đó đòi hỏi phải giải quyết các bài toán tối ưu hóa.

Để hiểu rõ thuật toán PSO hãy xem một ví dụ đơn giản về quá trình tìm kiếm thức ăn của một đàn chim. Không gian tìm kiếm thức ăn lúc này là toàn bộ không gian ba chiều mà chúng ta đang sinh sống. Tại thời điểm bắt đầu tìm kiếm cả đàn bay theo một hướng nào đó, có thể là rất ngẫu nhiên. Tuy nhiên sau một thời gian tìm kiếm một số cá thể trong đàn bắt đầu tìm ra được nơi có chứa thức ăn. Tùy theo số lượng thức ăn vừa tìm kiếm, mà cá thể gửi tín hiệu đến các các cá thể khác đang tìm kiếm ở vùng lân cận. Tín hiệu này lan truyền trên toàn quần thể. Dựa vào thông tin nhận được mỗi cá thể sẽ điều chỉnh hướng bay và vận tốc theo hướng về nơi có nhiều thức ăn nhất. Cơ chế truyền tin như vậy thường được xem

như là một kiểu hình của trí tuệ bầy đàn. Cơ chế này giúp cả đàn chim tìm ra nơi có nhiều thức ăn nhất trên không gian tìm kiếm vô cùng rộng lớn.

Như vậy đàn chim đã dùng trí tuệ, kiến thức và kinh nghiệm của cả đàn để nhanh chóng tìm ra nơi chứa thức ăn. Bây giờ chúng ta mô hình hóa mô hình sinh học này thường được gọi là quá trình phỏng sinh học mà chúng ta thường thấy trong các ngành khoa học khác. Một thuật toán được xây dựng dựa trên việc mô hình hóa các quá trình trong sinh học được gọi là thuật toán phỏng sinh học (bioinspired algorithms).

Hãy xét bài toán tối ưu của hàm số F trong không gian n chiều. Mỗi vị trí trong không gian là một điểm tọa độ n chiều. Hàm F là Hàm mục tiêu(fitness function) xác định trong không gian n chiều và nhận giá trị thực. Mục đích là tìm ra điểm cực tiểu của hàm F trong miền xác định nào đó. Ta bắt đầu xem xét sự liên hệ giữa bài toán tìm thức ăn với bài toán tìm cực tiểu của hàm theo cách như sau. Giả sử rằng số lượng thức ăn tại một vị trí tỉ lệ nghịch với giá trị của hàm F tại vị trí đó. Có nghĩa là ở một vị trí mà giá trị hàm F càng nhỏ thì số lượng thức ăn càng lớn. Việc tìm vùng chứa thức ăn nhiều nhất tương tự như việc tìm ra vùng chứa điểm cực tiểu của hàm F trên không gian tìm kiếm.

Bài toán trên chính là một bài toán PSO điển hình. Mục tiêu của chúng ta là áp dụng phương pháp tối ưu bầy đàn này vào hệ thống của chúng ta để tối ưu được sai số, tăng độ chính xác.

4.3.2 Áp dụng giải thuật Particle Swarm Optimization

Có 2 yếu tố chính để xây dựng nên một bài toán Particle Swarm Optimization chính là xác định được không gian tìm kiếm của bài toán và xác định được hàm mục tiêu của bài toán đó.

Theo như đã đề cập ở chương 3, sau khi giải được phương trình (3.10) ta sẽ có được kết quả gần đúng của bài toán, nhưng kết quả này không phải là kết quả tối ưu nhất. Tuy nhiên, ta có thể sử dụng kết quả này làm trung tâm của không gian, sau đó ta tìm kiếm xung quanh điển này với một bán kính cho trước (ví dụ 5m).

Việc tiếp theo, ta sẽ tìm hàm mục tiêu (Fitness Function) phù hợp với yêu cầu bài toán đặt ra. Ta nhận thấy, bản chất của bài toán chính là tối ưu được sai số của thiết bị so với vị trị thực tế. Vật hàm mục tiêu của chúng ta chính là sai số đạt được nhỏ nhất.

Xét hệ phương trình (3.3), như ta đã biết, các phép đo này là không chính xác bởi vì chịu tác động của các yếu tố ngoại cảnh như thời tiết, vật cản,… hoặc yếu tố nội cảnh như nguồn điện không ổn định, tính chính xác của antenna,… Tuy nhiên, ta sẽ sử dụng hệ phương trình này làm thước đo sai số cho bài toán.

Giả sử trong không gian tìm kiếm, ta có phần tử thứ k có vị trí $Z_k = [x_k\ y_k]$, khi đó ta có phương trình khoảng cách của phần tử này đến các điểm tham chiếu như sau:

$$\left|\hat{Z}_k - Zref_i\right| = \hat{d}_{ki} \tag{4.7}$$

Trong đó:

$\hat{Z}_k$ là vị trí của phần tử thứ k.

$Zref_i$ là vị trí của điểm tham chiếu thứ i ($i \in [1, n]$).

$\hat{d}_{ki}$Là khoảng cách của phần tử thứ k đến điểm tham chiếu thứ i.

Vậy sai số khoảng cách lúc này chính là:

$$f_k(\hat{Z}) = \Delta d_{ki} = \hat{d}_{ki} - d_i = \sqrt{(\hat{x}_k - x_i)^2 + (\hat{y}_k - y_i)^2} - d_i \tag{4.8}$$

Như vậy lúc này ta sẽ chọn hàm Fitness chính là tổng sai số của khoảng cách của phần tử k đối với các vị trí tham chiếu. Tuy nhiên, theo như công thức (4.8) Δd_{ki} có thể là một số âm hoặc dương, vì vậy để nhất quán, ta sẽ chọn hàm

Fitness chính là tổng bình phương sai số của khoảng cách của phần tử k đối với các vị trí tham chiếu. Cuối cùng ta có hàm Fitness như sau:

$$fitness\hat{Z} = \sum_{k=1}^{n} f_k^2(\hat{Z}) \tag{4.9}$$

Ngoài ra, một việc cũng quan trọng không kém đó chính là tổ chức quần thể. Việc tổ chức quần thể được biểu hiện qua việc khởi tạo xây dựng quần thể đó và phương pháp duy trì, phát triển quần thể.

Để khởi tạo xây dựng quần thể, ta cần biết các thông tin sau: trung tâm quần thể, bán kính không gian tìm kiếm và số lượng cá thể trong quần thể. Các thông tin này có thể được cài đặt trước bởi người dùng. Giả sử ta có trung tâm quần thể là $[x_{center}\ y_{center}]$, Bán kính không gian tìm kiếm là R, số lượng cá thể là N. Ta sẽ có các cá thể trong quần thể được sinh ra như sau:

$$\begin{cases} x_k = x_{center} + Dis_k * \sin(\ angle_k) \\ y_k = y_{center} + Dis_k * \cos(\ angle_k) \\ Dis_k = random(0, R) \\ angle_k = random(0, 2\pi) \end{cases} \tag{4.10}$$

Trong đó:

$[x_k\ y_k]$ là vị trí của phần tử thứ k.

$[x_{center}\ y_{center}]$ là vị trí của trung tâm quần thể

Dis_k là khoảng cách của phần tử thứ k đến trung tâm quần thể

$angle_k$ là góc của phần tử thứ k.

R là bán kính không gian quần thể.

Như vậy ta đã khởi tạo xây dựng xong quần thể. Việc tiếp theo của chúng ta là duy trì và phát triển quần thể.

Đầu tiên, để duy trì được quần thể thì mỗi cá thể sẽ được lặp đi lặp lại quá trình tìm kiếm để xác định được giải pháp tối ưu nhất. Trong mỗi vòng lặp, các phần tử sẽ tự theo dõi vị trí tốt nhất của chính nó , được xem như **Personal**

Best (pbest), giải pháp tốt ưu nhất của tất cả pbest chính là **Global Best (gbest).**

Global Best đóng vai trò quan trọng trong việc phát triển quần thể. Để dễ hiểu, ta ví dụ như có một đàn kiến, tản ra xung quanh để tìm kiếm thức ăn. Giả sử đàn kiến có một cách liên lạc tầm xa, thì mỗi khi một con kiến tìm được khu vực có thức ăn nhiều nhất mà nó tìm thất thì nó sẽ báo về lại đàn, lúc này khu vực có thức ăn mà con kiến này tìm được chính là Personal Best. Khu vực có nhiều thức ăn nhất trong tất cả khu vực mà các con kiến báo về chính là Global Best.

Mỗi khi khu vực có nhiều thức ăn nhất trong không gian tìm kiếm được cập nhật, thì cả đàn kiến sẽ có xu hướng tiến về khu vực đó để "khai thác". Trong lúc tiến về khu vực tốt nhất, trên đường đi có thể các con kiến sẽ kiếm được khu vực khác tốt hơn. Khi đó, Global Best được cập nhật, vậy là cả đàn có mục tiêu mới.

Quá trình lặp lại việc tìm kiếm Personal Best và Global Best chính là cơ sở để phát triển quần thể. Quá trình lặp lại có thể được biểu diễn như sau:

$$V_k(t+1) = \omega V_k(t) + c_1 r_1[pbest_k(t) - z_k(t)] + c_2 r_2[gbest(t) - z(t)] \quad (4.11)$$

$$Z_k(t+1) = Z_k(t) + V_k(t+1) \quad (4.12)$$

$$\omega = \omega_{max} - t\frac{\omega_{max} - \omega_{min}}{T} \quad (4.13)$$

Trong đó:

$V_k(t)$ là vận tốc của cá thể thứ k.

$Z_k(t)$ là vị trí hiện tại của phần tử thứ k (thuộc không gian tìm kiếm).

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.

Sau mỗi bước lặp tìm kiếm, ta sử dụng hàm Fitness để cập nhật lại Personal Best cũng như Global Best. Cuối cùng, sau khi kết thúc tìm hiểm, ta sẽ tìm được vị trí tối ưu nhất cho bài toán.

Đánh giá sơ lược bài toán, giả sử ta xây dựng một quần thể có bán kính 5m với 100 cá thể lúc ban đầu, nếu lặp 50 lần, trường hợp xấu nhất sẽ tạo ra 100*50 vị trí khác nhau. Vậy lúc này mỗi vị trí sẽ chiếm diện tích trung bình khoảng :

$$\frac{\pi * R^2}{n * T} = \frac{\pi * 5^2}{100 * 50} \approx 0.0157\ m^2$$

Lúc này sai số đạt được khoảng:

$$\Delta d \approx \frac{\sqrt{0.0157} * \sqrt{2}}{2} \approx 0.0886m = 8.86\ cm$$

Đây là sai số phụ thuộc vào các phép đo của thiết bị. Giả sử các thiết bị đo chính xác thì sai số đạt được là 8.86cm.