**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**.

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:

|  |  |  |
| --- | --- | --- |
|  |  | (4.1) |

Trong đó:

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

là ma trận mô hình chuyển đổi trạng thái.

là ma trận mô hình điều khiển đầu vào.

là vector điều khiển tại thời điểm .

vector ngẫu nhiên nhiễu hệ thống tại thời điểm .

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

|  |  |  |
| --- | --- | --- |
|  |  | (4.2) |

Trong đó:

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

là ma trận mô hình quan sát.

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 (). 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, và sẽ bằng 1. Vậy ta có thể viết lại phương trình (4.1) và (4.2) như sau:

|  |  |  |
| --- | --- | --- |
|  |  | (4.3) |
|  |  | (4.4) |

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

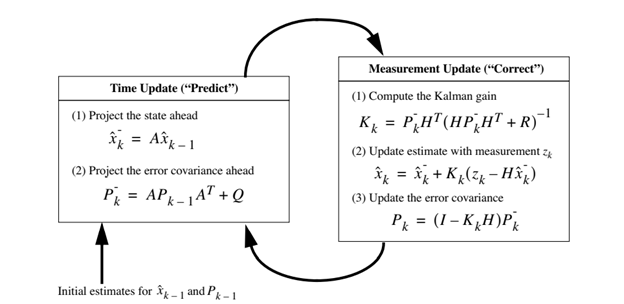
|  |  |  |
| --- | --- | --- |
|  |  | (4.5) |

Trong đó:

là hàm phân phối Gaussion.

là hiệp phương sai (covariance) nhiễu hệ thống.

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:

là hiệp phương sai ước lượng (estimate covariance).

là độ lợi Kalman (Kalman Gain)

Theo như hình 4.1, chúng ta cần khởi tạo và . Vậy ta phải cung cấp 5 tham số , , , Q và R để bộ lọc Kalman hoạt động.

ta khởi tạo bằng với dữ liệu RSSI đo được lần đầu tiên.

ta sẽ cho bằng 0, sẽ tự động thay đổi sau các bước tính.

ta khởi tạo bằng với , sẽ tự động thay đổi sau các bước tính.

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

ta lấy được bằng cách lấy dữ liệu RSSI thô và đánh giá sai số .

1. **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) |  |
| 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) |  |
| 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:

|  |  |  |
| --- | --- | --- |
|  |  | (4.6) |

Vậy với (RSSI tại 1m) và đã được cố định sẵn thì ta chỉ cần đo được 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.

1. **Particle Swarm Optimization**
   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.

* 1. **Áp dụng giải thuật Particle Swarm Optimization**