Table of Contents

Clear 1
Initialization 1
Loop 1
RSS Initializtion 2
First Algorithm 2
Second Algorithm 3
Third Algorithm 3
Fourth Algorithm 4
Plots 5

Clear

```
clc; % clear all
clear;
close all;
```

Initialization

```
R = 250; % initial parameters
L = 2 * R;
speed = 1;
sample_time = 0.1;
step_distance = speed * sample_time;
g = 150;
min_distance = sqrt(g);
max_distance = L - sqrt(g);
d1 = min_distance : step_distance : max_distance;
d2 = L - d1;
d3 = abs(R - d1);
d4 = abs(R - d1);
Ns = length(d1);
Pt = 20;
Po = 38;
grad1 = 2;
grad2 = 2;
alpha = exp(-1/85);
sigma1 = sqrt(8);
sigma2 = sqrt(sigma1^2 * (1 - alpha^2));
```

Loop

```
Repeats = 100 ; % number of iterations
Handoff_Matrix_by_algorithms = zeros(4,Repeats); % connected BS
PDF_location = zeros(4,Ns); % PDF location matrix

for j = 1 : Repeats
```

RSS Initializtion

calculate rss

```
    RSS01 = Pt - Po - (10 * grad1 * log10(d1) + 10 * grad2 * log10(d1/
g));
    RSS02 = Pt - Po - (10 * grad1 * log10(d2) + 10 * grad2 * log10(d2/
g));
    RSS_corner = Pt - Po - (10 * grad1 * log10(R) + 10 * grad2 *
 log10(R/g));
    RSS03 = RSS_corner - (10 * grad1 * log10(d3) + 10 * grad2 *
 log10(d3/g));
    RSS04 = RSS_corner - (10 * grad1 * log10(d4) + 10 * grad2 *
 log10(d4/g));
    for i=1:Ns
        if d3(i) < min_distance
            RSS03(i) = RSS_corner;
        end
        if d4(i) < min_distance
            RSS04(i) = RSS_corner;
        end
    end
    s1 = zeros(1,Ns);
    s2 = zeros(1,Ns);
    s3 = zeros(1,Ns);
    s4 = zeros(1,Ns);
    s1(1) = sigma1 * randn(1);
    s2(1) = sigma1 * randn(1);
    s3(1) = sigma1 * randn(1);
    s4(1) = sigma1 * randn(1);
    for i=2:Ns
        s1(i) = alpha * s1(i-1) + sigma2 * randn(1);
        s2(i) = alpha * s2(i-1) + sigma2 * randn(1);
        s3(i) = alpha * s3(i-1) + sigma2 * randn(1);
        s4(i) = alpha * s4(i-1) + sigma2 * randn(1);
    end
    RSS1 = RSS01 + s1;
    RSS2 = RSS02 + s2;
    RSS3 = RSS03 + s3;
    RSS4 = RSS04 + s4;


  %-----------------------------------------------------------------------
%
    % NEW_CODES %
```

First Algorithm

```
    RSS = [RSS1 ; RSS2 ; RSS3 ; RSS4]; % put all rsses in one matrix

    algorithm_1_which_BS = zeros(1,Ns); % algorithm one BS
    algorithm_1_which_BS(1) = 1 ; % initial BS
```

```
   for i = 2 : Ns
       [Max_RSS, Max_BS] = max(RSS(:,i)); % find maximum rss and
correlated bs
       if Max_BS == algorithm_1_which_BS(i-1)
           algorithm_1_which_BS(i) = algorithm_1_which_BS(i-1) ;
       else
           algorithm_1_which_BS(i) = Max_BS ; % change the rss
       end
   end

   % Number of Hand-offs in the First algorithm

   for i = 2 : Ns % calculate the number of hand-offs
       if algorithm_1_which_BS(i) ~= algorithm_1_which_BS(i-1)
           Handoff_Matrix_by_algorithms(1,j) =
Handoff_Matrix_by_algorithms(1,j) + 1 ;
       end
   end
```

Second Algorithm

```
   Threshold = -68 ;

   algorithm_2_which_BS = zeros(1,Ns); % algorithm two BS
   algorithm_2_which_BS(1) = 1 ; % initial BS

   for i = 2 : Ns
       if RSS(algorithm_2_which_BS(i-1),i-1) > Threshold % compare
with threshold
           algorithm_2_which_BS(i) = algorithm_2_which_BS(i-1) ;
       else
           [Max_RSS, Max_BS] = max(RSS(:,i));
           if Max_BS == algorithm_2_which_BS(i-1)
               algorithm_2_which_BS(i) = algorithm_2_which_BS(i-1) ;
           else
               algorithm_2_which_BS(i) = Max_BS ; % chnage the bss
           end
       end
   end

   % Number of Hand-offs in the Second algorithm

   for i = 2 : Ns % calculate the number of hand-offs
       if algorithm_2_which_BS(i) ~= algorithm_2_which_BS(i-1)
           Handoff_Matrix_by_algorithms(2,j) =
Handoff_Matrix_by_algorithms(2,j) + 1 ;
       end
   end
```

Third Algorithm

```
   H = 5 ; % in dbw
   algorithm_3_which_BS = zeros(1,Ns); % algorithm three BS
   algorithm_3_which_BS(1) = 1 ; % initial BS

   for i = 2 : Ns
       [Max_RSS, Max_BS] = max(RSS(:,i));
       if Max_BS == algorithm_3_which_BS(i-1)
           algorithm_3_which_BS(i) = algorithm_3_which_BS(i-1) ;
       else
           if RSS(algorithm_3_which_BS(i-1),i-1) + H > Max_RSS %
histersis comparison
               algorithm_3_which_BS(i) = algorithm_3_which_BS(i-1) ;
           else
               algorithm_3_which_BS(i) = Max_BS ; % change the bs
           end
       end
   end

   % Number of Hand-offs in the Third algorithm

   for i = 2 : Ns % calculate the number of hand-offs
       if algorithm_3_which_BS(i) ~= algorithm_3_which_BS(i-1)
           Handoff_Matrix_by_algorithms(3,j) =
Handoff_Matrix_by_algorithms(3,j) + 1 ;
       end
   end
```

Fourth Algorithm

```
   %Threshold = -68 ;
   %H = 5 ;

   algorithm_4_which_BS = zeros(1,Ns);
   algorithm_4_which_BS(1) = 1 ;

   for i = 2 : Ns
       if RSS(algorithm_4_which_BS(i-1),i-1) > Threshold % compare
with threshold
           algorithm_4_which_BS(i) = algorithm_4_which_BS(i-1) ;
       else
           [Max_RSS, Max_BS] = max(RSS(:,i));
           if Max_BS == algorithm_4_which_BS(i-1)
               algorithm_4_which_BS(i) = algorithm_4_which_BS(i-1) ;
           else
               if RSS(algorithm_4_which_BS(i-1),i-1) + H > Max_RSS %
compare using histersis
                   algorithm_4_which_BS(i) =
algorithm_4_which_BS(i-1) ;
               else
                   algorithm_4_which_BS(i) = Max_BS ;
               end
           end
       end
```

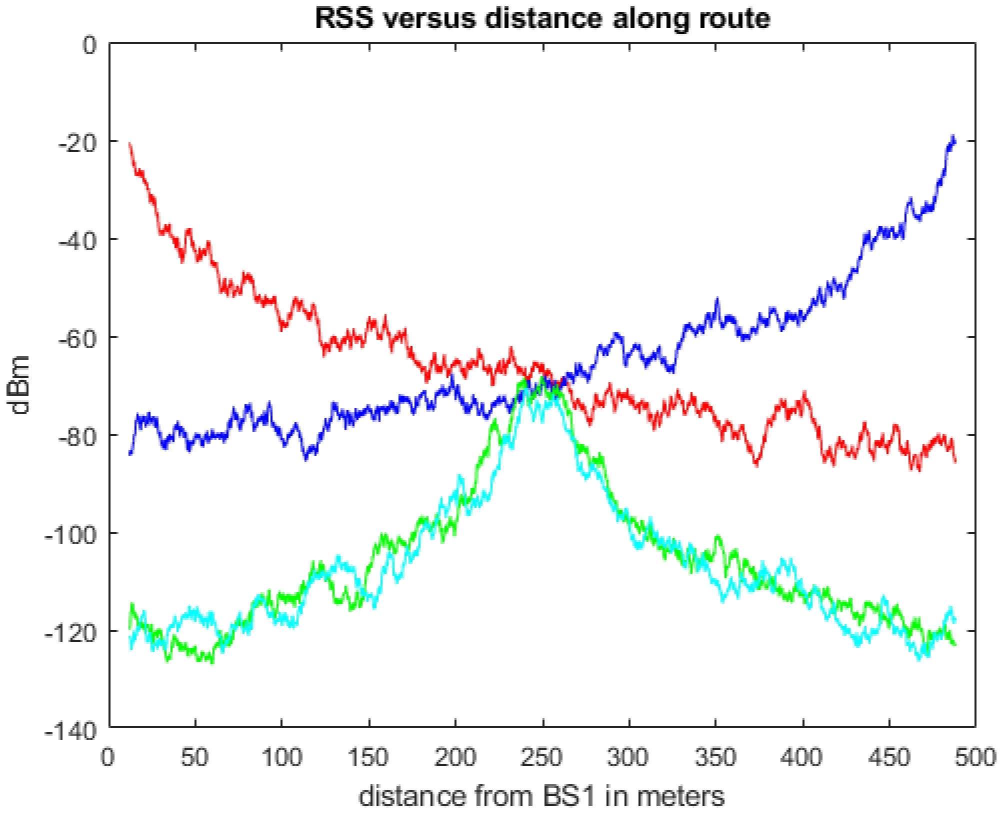
```
    end

    % Number of Hand-offs in the Fourth algorithm

    for i = 2 : Ns % calculate the number of hand-offs
        if algorithm_4_which_BS(i) ~= algorithm_4_which_BS(i-1)
            Handoff_Matrix_by_algorithms(4,j) =
 Handoff_Matrix_by_algorithms(4,j) + 1 ;
        end
    end


  %-------------------------------------------------------------------------
%
    % PDF of Location of Hand-offs

    which_BS = [algorithm_1_which_BS; algorithm_2_which_BS; ...
        algorithm_3_which_BS; algorithm_4_which_BS];

    for i1 = 1 : 4 % find the pdf location of hand-offs
        for j1 = 2 : Ns
            if which_BS(i1,j1) ~= which_BS(i1,j1-1)
                PDF_location(i1,j1) = PDF_location(i1,j1) + 1;
            end
        end
    end

end

%-------------------------------------------------------------------------
%
% PDF of Number of Hand-offs
n = 1 : 100;
PDF_number = zeros(4,100);

for i = 1 : 4
    for j = 1 : 100
        for k = 1 : Repeats
            if Handoff_Matrix_by_algorithms(i,k) == j
                PDF_number(i,j) = PDF_number(i,j) + 1;
            end
        end
    end
end
```

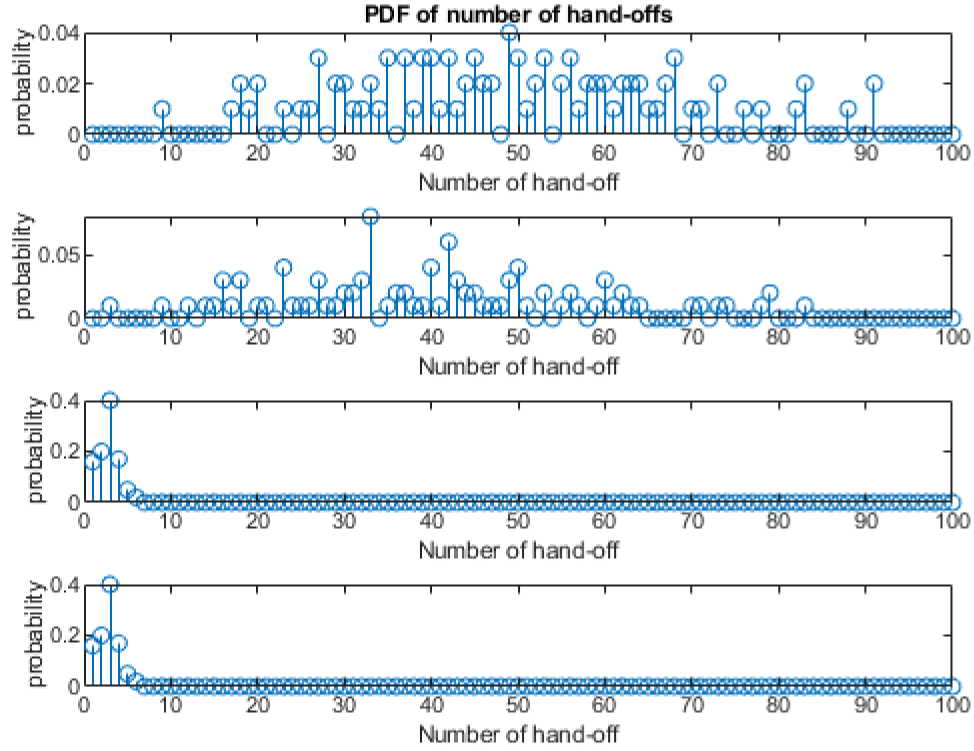
Plots

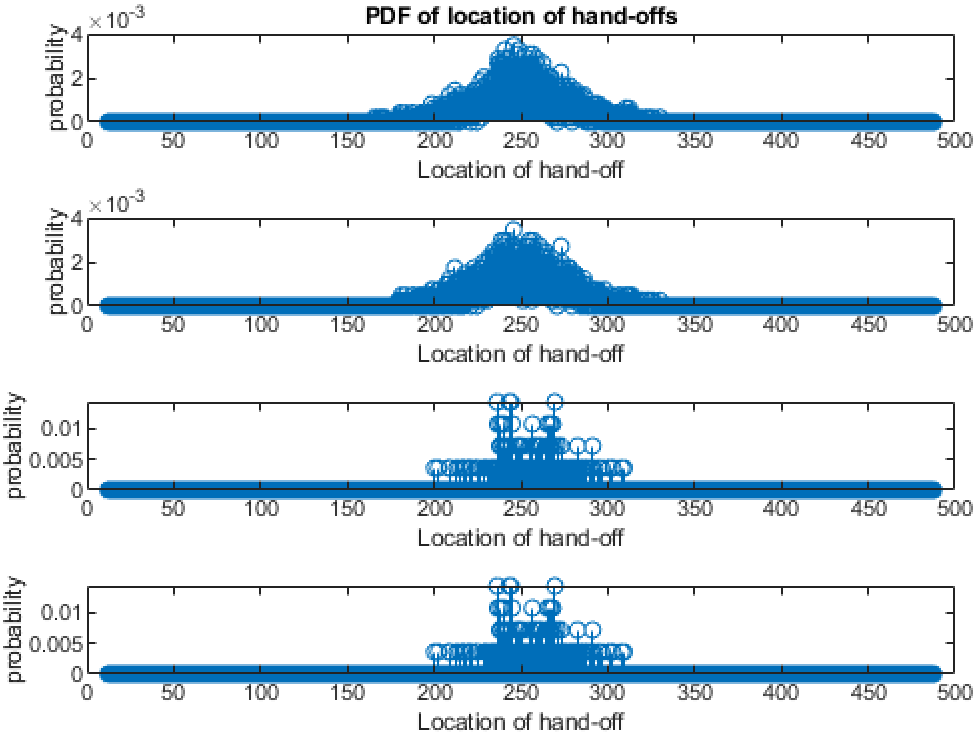
```
% Plot the RSS values obtained
figure(1)
plot(d1, RSS1,'r')
hold on
plot(d1, RSS2,'b')
hold on
```

```
plot(d1, RSS3,'g')
hold on
plot(d1, RSS4,'c')
title('RSS versus distance along route')
xlabel('distance from BS1 in meters');
ylabel('dBm');

% PDF of number of hand-offs
figure(2)

subplot(4,1,1);
stem(n,PDF_number(1,:)./Repeats);
title('PDF of number of hand-offs')
xlabel('Number of hand-off');
ylabel('probability');

subplot(4,1,2);
stem(n,PDF_number(2,:)./Repeats);
xlabel('Number of hand-off');
ylabel('probability');

subplot(4,1,3);
stem(n,PDF_number(3,:)./Repeats);
xlabel('Number of hand-off');
ylabel('probability');

subplot(4,1,4);
stem(n,PDF_number(4,:)./Repeats);
xlabel('Number of hand-off');
ylabel('probability');

% PDF of locations of hand-offs
figure(3)

subplot(4,1,1);
stem(d1,PDF_location(1,:)./sum(Handoff_Matrix_by_algorithms(1,:)));
title('PDF of location of hand-offs')
xlabel('Location of hand-off');
ylabel('probability');

subplot(4,1,2);
stem(d1,PDF_location(2,:)./sum(Handoff_Matrix_by_algorithms(2,:)));
xlabel('Location of hand-off');
ylabel('probability');

subplot(4,1,3);
stem(d1,PDF_location(3,:)./sum(Handoff_Matrix_by_algorithms(3,:)));
xlabel('Location of hand-off');
ylabel('probability');

subplot(4,1,4);
stem(d1,PDF_location(4,:)./sum(Handoff_Matrix_by_algorithms(4,:)));
xlabel('Location of hand-off');
ylabel('probability');
```

```
%clearvars -except algorithm_1_which_BS algorithm_2_which_BS
 algorithm_3_which_BS ...
    %algorithm_4_which_BS Handoff_Matrix_by_algorithms RSS d1 n PDF Ns
 PDF_number...
    %PDF_location ;
```




PDF of number of hand-offs
probability
0
0.02
0.04
0.05
0.2
0.4
0
10
20
30
40
50
60
70
80
90
100
Number of hand-off


PDF of location of hand-offs
probability
$\times 10^{-3}$
0
2
4
0.005
0.01
0
50
100
150
200
250
300
350
400
450
500
Location of hand-off

Published with MATLAB® R2020b