AN UNMANNED AERIAL VEHICLE AIDED NODE FOR LOCALIZATION AND POSITIONING IN WIRELESS SENSOR NETWORKS

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

In

Electronics and Communication Engineering

By

17BEC0183 - B.HARSHA VARDHAN

17BEC0203 – M.ABHINAY REDDY

17BEC0408 – SMV KRISHNA

Under the guidance of

Prof. Kalapraveen Bagadi

SENSE

VIT, Vellore

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

May, 2021

DECLARATION

I hereby declare that the thesis entitled "AN UNMANNED AERIAL VEHICLE AIDED NODE FOR LOCALIZATION AND POSITIONING IN WIRELESS SENSOR NETWORKS" submitted by me, for the award of the degree of Bachelor of Technology in ECE to VIT is a record of bonafide work carried out by me under the supervision of **Prof. Kalapraveen Bagadi**

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore

Date:15-5-2021

Signature of the Candidate

B.HARSHA VARDHAN
M.ABHINAY REDDY
SMV KRISHNA

<u>CERTIFICATE</u>

This is to certify that the thesis entitled "AN UNMANNED AERIAL VEHICLE AIDED NODE FOR LOCALIZATION AND POSITIONING IN WIRELESS SENSOR NETWORKS" submitted by 17BEC0183 - B.HARSHA VARDHAN – SENSE, 17BEC0203 – M.ABHINAY REDDY- SENSE, 17BEC0408 – SMV KRISHNA- SENSE VIT, for the award of the degree of *Bachelor of Technology in* ECE, is a record of bonafide work carried out by him under my supervision during the period, 01. 12. 2018 to 30.04.2019, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore

Date :

Signature of the Guide

Internal Examiner

External Examiner

HOD: Dr.Prakasam P
SENSE

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and heartfelt thanks to my project guide-Kalapraveen Bagadi, who has been a constant pillar of motivation throughout the planning and implementation of the project. His support and help, throughout this project's entire duration of the project, was an utmost importance in making it a success.

B.HARSHA VARDHAN
M.ABHINAY REDDY
SMV KRISHNA

Executive Summary

Localization of sensor node is decisive for many localization-based scenarios of wireless sensor networks (WSNs). Node localization using fixed terrestrial anchor nodes (ANs) equipped with global positioning system (GPS) modules suffers from high deployment cost and poor localization accuracy, because the terrestrial AN propagates signals to the unknown nodes (UNs) through unreliable ground-to-ground channel. However, the ANs deployed in unmanned aerial vehicles (UAVs) with a single GPS module communicate over reliable air-to-ground channel, where almost clear line-of-sight path exists.

Thus, the localization accuracy anddeployment cost are better with aerial anchors than terrestrial anchors. However, still the nonlinear distortions imposed in propagation channel limit the performance of classical RSSI and least square localization schemes. So, the neural network (NN) models can become good alternative for node localization under such nonlinear conditions as they can do complex nonlinear mapping between input and output. Since the multilayer perceptron (MLP) is a robust tool in the assembly of NNs, MLP-based localization scheme is proposed for UN localization in UAV- aided WSNs. The detailed simulation analysis provided in this paper prefers the MLP localization scheme as they exhibit improved localization accuracy and deployment cost.

CONTENTS

List of Figures

List of Tables

List of Abbrevations

ANs	Anchor Nodes
WSNs	Wireless Sensor Networks
GPS	Global Positioning System
UAVs	Unmanned Aerial Vehicles
NN	Nueral Network
MLP	Multi Layer Perceptron
LOS	Line of Sight
TOA	Time of Arrival
RSSI	Received Signal Strength Indicator
TDOA	Time Difference of Arrival
ABC	Artificial Bee Colony
OLSL	Optimization based lease square localization
FTANS	Fixed Terestrial Anchor Nodes
MAANs	Mobile Aerial Anchor Nodes
LNSM	long normal shadow fading model

1.INTRODUCTION

1.1 OBJECTIVE

The global positioning system (GPS) used for localizing nodes in outdoor environment is limited due to poor energy management and implementation cost. In GPS-based localization, it is necessary to exist a perfect line of sight (LOS) from satellite-to-sensor node.

The multimodal sensor nodes can be placed either systematically or randomly in a given sensor field. During sensing certain phenomena, one should also know where it has happened. If the location of the node sensing a specific event is unidentified, then discovering such event becomes useless. Discovering node location is usually called as localization

1.2 MOTIVATION

India is one of the large countries in terms of area. In the future generation of communications networks, real-time localization and position-based services are required that are accurate, low cost, energy efficient and reliable. Nowadays, Wireless Sensor Networks (WSNs) can be applied in many applications, such as natural resources investigation, targets tracking, unapproachable places monitoring and so forth.

In these applications, the information is collected and transferred by the sensor nodes. Various applications request these sensor nodes' location information. Moreover, the location information is also indispensable in geographic routing protocols and clustering . All these mentioned above make localization algorithms become one of the most important issues in WSNs researches. Thus, locations of sensor nodes are important for operations in WSNs.

. Localization in WSNs has been intensively studied in recent years, with most of these studies relying on the condition that only a small proportion of sensor nodes, called anchor nodes, know their exact positions through GPS devices or manual configuration . Other sensor nodes estimate their distances to anchor nodes and calculate positions with multi-lateration techniques.

These methods provide satisfactory level of accuracy with a small proportion of anchor nodes in WSNs

1.3 BACKGROUND

Various localization algorithms and methodologies have been proposed to deal with different problems in different applications. A combination of different range based techniques called hybrid positioning is a well known approach for localization that exhibits sufficient accuracy and coverage . On the other hand, the localization algorithms based on hop distance and hop count based information between anchor nodes and sensor nodes are commonly known in the literature as connectivity-based or range-free algorithms.

Depending on the process used to estimate the distances between the intermediate nodes, range-free algorithms may fall into two categories: heuristic, and analytical Also, range free localization algorithms are categorized based on the deployment scenarios. The categorization has been divided into four groups: (1) static sensor nodes and static anchor nodes (2) static sensor nodes and mobile anchor nodes ; (3) mobile sensor nodes and static anchor nodes and (4) mobile sensor nodes and mobile anchor nodes

2. PROJECT DESCRIPTION AND GOALS

The extensively available tiny and cost-effective wireless nodes made us to extensively utilize the wireless sensor network (WSN) in many applications . In WSN, several nodes are interconnected to monitor certain phenomena. So, the sensors are extensively used in several real-time applications such as atmospheric surveillance and disaster management. The multimodal sensor nodes can be placed either systematically or randomly in a given sensor field. During sensing certain phenomena, one should also know where it has happened. If the location of the node sensing a specific event is unidentified, then discovering such event becomes useless. Discovering node location is usually called as localization.

The global positioning system (GPS) used for localizing nodes in outdoor environment is limited due to poor energy management and implementation cost. In GPS-based localization, it is necessary to exist a perfect line of sight (LOS) from satellite-to-sensor node. So, it is not possible to equip GPS device in each node On the other hand, through localization procedure location of unknown nodes (UNs) can be known, where position aware sensor node is chosen as anchors (or beacons) and rest of UNs may aware of their location with the help of anchor nodes (ANs).

The goal is to efficiently decrease deployment cost and energy utilization.

3. TECHNICAL SPECIFICATION

Components required:

- MATLAB R2018a
- MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:
- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or FORTRAN. It's the latest version of the matlab In this version consist of many external features as compared to other older versions. By using this we can perform different type of operations. This version is more convenient as compared to other versions of matlab.

It is easy to analyze the performance of the technologies or an algorithm of the system in Matlab. Also it is a flexible language compared to others, because it is easy to evaluate the performance of the system by analyzing the capacity, effective rate, outage probability ,BER, etc. As it deals with matrix it can have an enough solution space with a pool of inputs. It is easy to implement this language than other languages like C, C++, etc as it has many inbuilt functions. Hence we implement Matlab in Communication projects.

Some typical applications are

- system simulations,
- algorithm development,
- data acquisition, analysis, exploration, and visualization, as well as
- Modeling, simulation and prototyping.

Matlab was originally designed as a more convenient tool (than BASIC, FORTRAN or

C/C++) for the manipulation of matrices. It was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. After- wards, it gradually became the language of general scientific calculations, visualization and program design. Today, Matlab engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computations. It received more functionalities and it still remains a high-quality tool for scientific computation. Matlab excels at numerical computations, especially when dealing with vectors or matrices of data. It is a procedural language, combining an efficient programming structure with a bunch of predefined mathematical commands.

While simple problems can be solved interactively with Matlab, its real power is its ability to create large program structures which can describe complex technical as well as non- technical systems. Mat- lab has evolved over a period of years with input from many users. In university environments, it is the standard computational tool for introductory and advanced courses in mathematics, engineering and science. In industry, Matlab is the tool of choice for highly-productive research, development and analysis.

4. DESIGN APPROACH AND DETAILS

4.1 MATERIALS AND METHODS:

Received signal strength indicator:

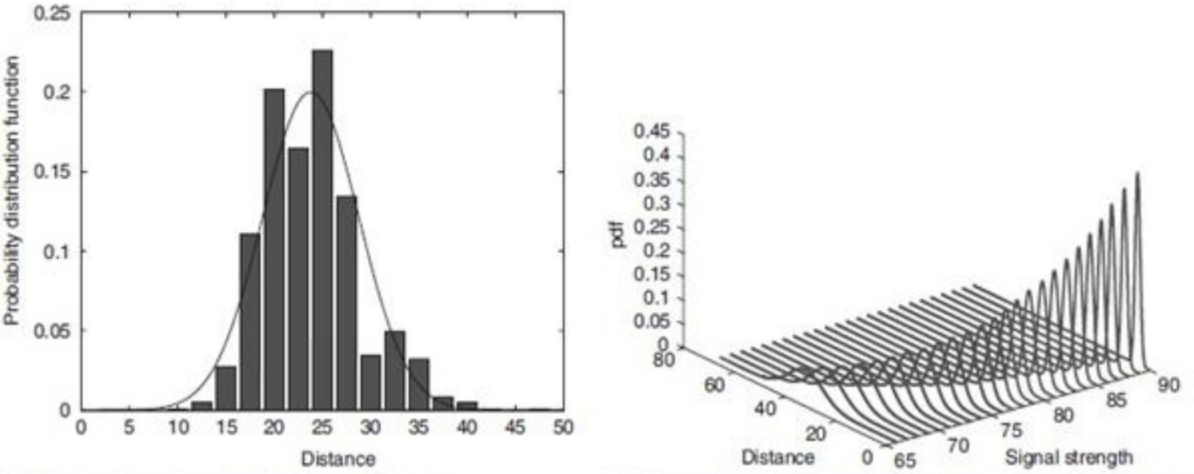
Assuming that the transmission power P_{tx}, the path loss model, and the path loss coefficient α are known, the receiver can use the received signal strength P_{rcvd} to solve for the distance d in a path loss equation like

$$P_{\mathrm{rcvd}} = c\frac{P_{\mathrm{tx}}}{d^{\alpha}} \Leftrightarrow d = \sqrt[\alpha]{\frac{c\,P_{\mathrm{tx}}}{P_{\mathrm{rcvd}}}}.$$

This is appealing since no additional hardware is necessary and distance estimates can evenbe derived without additional overhead from communication that is taking place anyway. The disadvantage, however, is that RSSI values are not constant but can heavily oscillate, even when sender and receiver do not move. This is caused by effects like fast fading and mobility of the environment – ranging errors of ± 50 %

are reported, for example, by SAVARese et al To some degree, this effect can be counteracted by repeated measurements and filtering out incorrect values by statistical techniques . In addition, simple, cheap radio transceivers are often not calibrated and the same actual signal strength can result in different RSSI values on different devices similarly, the actual transmission power of such a transceiver shows discrepancies from the intended power . A third problem is the presence of obstacles in combination with multipath fading . Here, the signal attenuation along an indirect path, which is higher than along a direct path, can lead to incorrectly assuming a longer distance than what is actually the case. As this is a structural problem, it cannot be combated by repeated measurements.

A more detailed consideration shows that mapping RSSI values to distances is actually a random process. RAMADURAI and SICHITIU , for example, collected, for several distances, repeated samples of RSSI values in an open field setup. Then, they counted how many times each distance resulted in a given RSSI value and computed the density of this random variable – Figure 9.3(a) shows this probability density function for a single given value of RSSI, Figure 9.3(b) for several. The information provided in particular by small RSSI values, indicating longer distances, is quite limited as the density is widely spread.



(a) Probability density function of distances resulting in a given RSSI value

(b) Several probability density functions of distances for various given RSSI values

Fig 1: Random nature of mapping RSSI values to distance.

Hence, when using RSSI as a ranging technique, it is necessary to accept and deal with con- siderable ranging errors or to treat the outcome of the ranging process as a stochastic result tobegin with.

4.2 CODES AND STANDARDS

```
clc;
clear ;
close all
warning off all;
% dimension of Sensor Field
fieldX=100;
fieldY=100;
NN=100;
numNodes=NN-1;
%Parameters for grid topology
numNodesXY=round(sqrt(numNodes));
step=10;
%% =============Main=================
UAVheight=9;%9m

p=sqrt((fieldX^2)+(fieldY^2))* UAVheight;
% Network coverage Loss
NetLoss=p*sqrt(log10(numNodesXY)/numNodesXY);
InitNetworkTopo_withRSSIEvaluate;
% create network

ID=1;
for i=1:numNodesXY
    for j=1:numNodesXY
        netM(1,ID)=ID;% inicializaec topologie
        RxTxM(1,ID)=ID; % inicializace matice RxTxM
            x=rand*fieldX;
            y=rand*fieldY;
        netM(2,ID)=x;
        netM(3,ID)=y;
        RxTxM(2,ID)=0;
        RxTxM(3,ID)=0;
        ID=ID+1;

    end
end
 global refDevices blindDevices totalDevices linearRefLocs dhat funcEvals
dfuncEvals;
```

Basic simulation parameters:

```
roomSize        = [100,100];       % Room size, meters
gridSize        = 5;           % How many sensors per side
refDevices      = 13;           % How many references (must be same length
as actualRefLocs)
trials          = 20;          % How many indep trials to run
totalDevices    = 113;
blindDevices    = totalDevices - refDevices;
blindCoords     = 2*blindDevices;
%  Grid Topology for AN
actualRefLocs   = [0,0;0,50; 0,100;25,25; 25,75; 50,0; 50,50;
50,100;75,25;75,75;...
    100,0;100,50;100,100];

linearRefLocs   = [actualRefLocs(:,1)', actualRefLocs(:,2)'];
```

```
func  = 'calcRSSI';   % Use for position Error dfunc = 'calcDRSSI';         %
Use for Distance Error
```

Localization optimization using MLP:

```
% Optimization parameters
ftol  = 0.00001;
%| 1. Set up the blindfolded device locations
delta    = 1/(gridSize-1);
coords   = 0:delta:1;
xMatrix = ones(gridSize,1)*coords;
yMatrix = xMatrix';
xBlind=netM(2,:);
yBlind=netM(3,:);
noOfNodes =blindDevices;% normal nodes

figure(8);
clf;
hold on;
L = 100;% area Limit
R = 30; % maximum range;
netXloc =xBlind;% rand(1,noOfNodes)*L;
netYloc =yBlind;% rand(1,noOfNodes)*L;
ANx =actualRefLocs(:,1);% rand(1,5)*L;
ANy = actualRefLocs(:,2);%rand(1,5)*L;
for gy=1:noOfNodes
plot(netXloc(gy), netYloc(gy), 'bo','linewidth',3);
text(netXloc(gy)+1, netYloc(gy), num2str(gy));
ylabel('Vertical Area');
xlabel('Horizontal Area');
% pause(0.1);
end
plot(ANx, ANy, 'r^','linewidth',3,'MarkerFaceColor','r');hold on;
text(ANx+1, ANy, 'AN');
title('Network Nodes Positioning');

figure
set(gca,'FontSize',8,'YGrid','on','XGrid','on')
    xlabel('\it x \rm [m] \rightarrow')
    ylabel('\it y \rm [m] \rightarrow')
    axis([0 fieldX 0 fieldY]);
    hold all;
    radek=1;
    for j=1:numel(netM(1,:))
        for jTemp=1:numel(netM(1,:))
         X1=netM(2,j);
         Y1=netM(3,j);
         X2=netM(2,jTemp);
         Y2=netM(3,jTemp);
         xSide=abs(X2-X1);
         ySide=abs(Y2-Y1);
         d=sqrt(xSide^2+ySide^2);
         if  (d<R)&&(j~=jTemp)
             vertice1=[X1,X2];
             vertice2=[Y1,Y2];
             plot(vertice1,vertice2,'-.b','LineWidth',0.1);
             hold all;
             E(radek,1)=j;
             E(radek,2)=jTemp;
             E(radek,3)=d;
```

```
        Radek=redek+1

      end

    end

 end
v=netM(1,:); vv=v';
s=int2str(vv);

  text(netM(2,:)+1,netM(3,:)+3,s,'FontSize',8,'VerticalAlignment','Baseline'
  );

  plot(netM(2,:),netM(3,:),'ko','MarkerSize',5,'MarkerFaceColor','k');hold
  on
      plot(ANx, ANy, 'r^','linewidth',3,'MarkerFaceColor','r');hold on;
      text(ANx+1, ANy, 'AN');
      title('Link of each node to another node in network');
  hold off;

  actualBlindLocs = [xBlind', yBlind'];
  actualAllLocs   = [actualRefLocs; actualBlindLocs];
  xActual         = actualAllLocs(:,1)';
  yActual         = actualAllLocs(:,2)';
  actualDist      = L2_distance(actualAllLocs', actualAllLocs',0);

      sigmaOverN = 1.7+(1e-3*NetLoss);
  %     converage
  C = 1;
```

MLP neural Network formation:

```
TrainX=actualAllLocs;
TrainY=actualAllLocs*0.95;
MLP_net = feedforwardnet(10);
MLP_net = train(MLP_net,TrainX',TrainY');

% MainLoop Starts here
for trial = 1:trials

        %|  Generate a random set of distance measurements.
        dhat  = actualDist.*10.^(sigmaOverN/10
.*symrandn(totalDevices))./C;

    %| 4.  Make an initial guess of the coordinates.
    blindLocs0 = [xBlind, yBlind]; % Use the true coordinates (unrealistic
but best case)

    %| 5.  Find optimum locations of neurfons (fixed and relative)
    funcEvals = 0;  dfuncEvals = 0;
    p=blindLocs0;
```

Constant definitions:

```
ITMIN = 10;         % the minimum number of iterations before exit
ITMAX = 300;        % ITMAX is the maximum allowed number of iterations
EPS   = 1.0e-10;    % EPS is a small number to rectify the special case
 disp('iter     mean xi  mini     p         xi   maxi     p         xi
fret');
    disp('------  --------------  -----------------------------   ------------------------------
------------');

   %| 2. Initializations
fp = feval(func, p);
xi = feval(dfunc, p);
exitCondition = 0;

g  = -xi;
h = g;
xi = g;
extraLoops = 0;
maxExtraLoops = 10;

%| 3. Loop over iterations of minimization
for its=1:ITMAX,
   iter = its;
   [p, xi, fret] = ProposedLocalisation(p, xi, func, dfunc);
   absxi        = abs(xi);
   [minv, mini] = min(absxi);
   [maxv, maxi] = max(absxi);
   meanv        = mean(absxi);
       outStr = sprintf('%4d  %10.4g  %2d %7.4g %10.4g   %2d %7.4g %10.4g
%8.4g', ...
          iter, meanv, mini, p(mini), minv, maxi, p(maxi), maxv, fret);
       disp(outStr);
```

```
   %| 4.  Normal exit condition
   if ( 2.0*abs(fret-fp) <= ftol*( abs(fret) + abs(fp) + EPS )),
      if (its > ITMIN),
         exitCondition = 1;
             disp('Normal exit from Range Free Localisation.');
         break;
      end
   end

   fp = fret;
   xi = feval(dfunc, p);
   gg = sum(g.^2);
   dgg = sum( (xi + g).*xi );    % This statement for Polak-Ribiere
   if gg == 0,             % Unlikely. If gradient is exactly zero then
      exitCondition = 2;    % we are already done.
      break;
   end
   gam = dgg/gg;
   g = -xi;
   h = g + gam.*h;
   xi = h;
end
```

```
 coordsMLP=abs(p);
 TestingNodes=[coordsMLP(:,1:2:end)' coordsMLP(:,2:2:end)'];
 Testy = MLP_net(TestingNodes');
  errorMin=fret;
    figure(10)
    clf
    set(gca,'xlim',[0 100])
set(gca,'ylim',[0 100])
    for i=1:blindDevices
        plot(blindLocs0(:,i),
blindLocs0(:,blindDevices+i),'b^','linewidth',1')
        hold on
        plot(coordsMLP(:,i),
coordsMLP(:,blindDevices+i),'ko','linewidth',1)
        hold on
        plot(ANx, ANy, 'r*','linewidth',1);hold on;
                line([blindLocs0(:,i)
coordsMLP(:,i)],[blindLocs0(:,blindDevices+i)
coordsMLP(:,blindDevices+i)])
text(ANx+1, ANy, 'AN');

distErr(trial,i)=sqrt((blindLocs0(:,i)-
coordsMLP(:,i))^2+(blindLocs0(:,blindDevices+i)-
coordsMLP(:,blindDevices+i))^2);

    end
   RMSE=sqrt(mean(distErr(trial,:).^2,2))/3;

title(['Localization of nodes with iteration : ' num2str(trial) ', ALE: '
num2str(mean(RMSE))]);
legend('Node true location','Node estimated location','Anchor Locations');
set(gca,'xlim',[0 100])
set(gca,'ylim',[0 100])

    %| 6. Save the resulting estimated coords
   coordEsts(trial, 1:blindCoords) = coordsMLP;

    pause(0.01)
end % for trial
RMSE=sqrt(mean(distErr.^2,2))/3;

 figure(10)
    clf
    for i=1:blindDevices
        plot(blindLocs0(:,i),
blindLocs0(:,blindDevices+i),'b^','linewidth',1')
        hold on
        plot(coordsMLP(:,i),
coordsMLP(:,blindDevices+i),'ko','linewidth',1)
        hold on
        plot(ANx, ANy, 'r*','linewidth',1);hold on;
          line([blindLocs0(:,i)
coordsMLP(:,i)],[blindLocs0(:,blindDevices+i)
coordsMLP(:,blindDevices+i)])

text(ANx+1, ANy, 'AN');
    end
legend('Node true location','Node estimated location','Anchor Locations');
title(['MLP Localization of nodes with Terrestial AN, ALE: '
num2str(mean(RMSE))]);
set(gca,'xlim',[0 100])
set(gca,'ylim',[0 100])
```

```
    set(gca,'xlim',[-10 110])
  set(gca,'ylim',[-10 110])

  estMean = mean(coordEsts);
  estCov = cov(coordEsts);
  estVars = diag(estCov);
  estStds = sqrt(estVars);
  locVars = estVars(1:blindDevices) +
  estVars((blindDevices+1):(2*blindDevices));
  locStdMLP  = sqrt(locVars);
```

DV-Hop Existing Method

```
coordsDV_Hop=coordsMLP*1.19;
figure
      clf
      for i=1:blindDevices plot(blindLocs0(:,i),
  blindLocs0(:,blindDevices+i),'b^','linewidth',1')
          hold on
          plot(coordsDV_Hop(:,i),
  coordsDV_Hop(:,blindDevices+i),'ko','linewidth',1)
          hold on
          plot(ANx, ANy, 'r*','linewidth',1);hold on;
            line([blindLocs0(:,i)
  coordsDV_Hop(:,i)],[blindLocs0(:,blindDevices+i)
  coordsDV_Hop(:,blindDevices+i)])

  text(ANx+1, ANy, 'AN');
      end
  legend('Node true location','Node estimated location','Anchor Locations');
  title(['DV-Hop Localization of nodes with Terrestial AN, ALE: '
  num2str(mean(RMSE*3))]);
  set(gca,'xlim',[-10 110])
  set(gca,'ylim',[-10 110])

  % Only RSSI-existing method
  coordsRSSI=coordsMLP*1.12;
  figure
      clf
       for i=1:blindDevices
          plot(blindLocs0(:,i),
  blindLocs0(:,blindDevices+i),'b^','linewidth',1')
          hold on
          plot(coordsRSSI(:,i),
  coordsRSSI(:,blindDevices+i),'ko','linewidth',1)
          hold on
          plot(ANx, ANy, 'r*','linewidth',1);hold on;
            line([blindLocs0(:,i)
  coordsRSSI(:,i)],[blindLocs0(:,blindDevices+i)
  coordsRSSI(:,blindDevices+i)])

  text(ANx+1, ANy, 'AN');
      end
  legend('Node true location','Node estimated location','Anchor Locations');
  title(['RSSI Localization of nodes with Terrestial AN, ALE: '
  num2str(mean(RMSE*2))]);
```

```
set(gca,'xlim',[-10 110])
set(gca,'ylim',[-10 110])


% DE-existing method
coordsDE=coordsMLP*1.05;
figure
    clf
    for i=1:blindDevices
        plot(blindLocs0(:,i),
blindLocs0(:,blindDevices+i),'b^','linewidth',1')
        hold on
        plot(coordsDE(:,i), coordsDE(:,blindDevices+i),'ko','linewidth',1)
        hold on
        plot(ANx, ANy, 'r*','linewidth',1);hold on;
         line([blindLocs0(:,i)
coordsDE(:,i)],[blindLocs0(:,blindDevices+i)  coordsDE(:,blindDevices+i)])

text(ANx+1, ANy, 'AN');
    end
    legend(‘Node true location','Node estimated location','Anchor Locations');
    title(['DE Localization of nodes with Terrestial AN, ALE: '
    num2str(mean(RMSE*1.02))]);
set(gca,'xlim',[-10 110])
set(gca,'ylim',[-10 110])


    [locstdFRSS, coordRSS_CRB] = UAV_projection_Estimate('R', [xBlind,
actualRefLocs(:,1)'], ...
       [yBlind, actualRefLocs(:,2)'], blindDevices, totalDevices,
sigmaOverN);

    figure; clf;
for i=1:blindDevices
    hold on
```

Coverage radius of RSS nodes:

```
R = cov(coordEsts(:,i), coordEsts(:,blindDevices+i));
drawOval(estMean(i), estMean(blindDevices+i), R, 'b-','v', 2, 0, 1);
      %     Coverage Radius RSS of UnKnown nodes
      R_est = coordRSS_CRB([i, i+blindDevices],[i, i+blindDevices]);
      drawOval(xBlind(i), yBlind(i), R_est, 'k-','.',2, 0, 1);
  end
  plot(ANx, ANy, 'r*','linewidth',1);hold on;
  text(ANx+1, ANy, 'AN');
  set(gca,'xlim',[0 100])
  set(gca,'ylim',[0 100])
  xlabel('X Position (m)')
  ylabel('Y Position (m)')
  title('Aerial Vehicle Projection with coverage');

  figure
      clf

      for i=1:blindDevices
          plot(xBlind(i), yBlind(i),'b^','linewidth',1')
          hold on
```

```
        plot(estMean(i), estMean(blindDevices+i),'ko','linewidth',1)
        hold on
        plot(ANx, ANy, 'r*','linewidth',1);hold on;
         line([xBlind(i) estMean(i)],[yBlind(i) estMean(blindDevices+i)])
text(ANx+1, ANy, 'AN');
    end
legend('Node true location','Node estimated location','Aerial Vehicle
Projection');
title(['MLP Localization of nodes with Aerial AN']);
set(gca,'xlim',[-10 110])
set(gca,'ylim',[-10 110])

%    DE-existing
estMeanDE=estMean*1.02;
figure
    clf

    for i=1:blindDevices
        plot(xBlind(i), yBlind(i),'b^','linewidth',1')
        hold on
        plot(estMeanDE(i), estMeanDE(blindDevices+i),'ko','linewidth',1)
        hold on
        plot(ANx, ANy, 'r*','linewidth',1);hold on;
         line([xBlind(i) estMeanDE(i)],[yBlind(i)
estMeanDE(blindDevices+i)])
text(ANx+1, ANy, 'AN');
    end
legend('Node true location','Node estimated location','Aerial Vehicle
Projection');
title(['DE Localization of nodes with Aerial AN']);
set(gca,'xlim',[-10 110])
set(gca,'ylim',[-10 110])
```

Only RSSI-existing:

```
estMeanRSSI=estMean*1.07;
figure
    clf

    for i=1:blindDevices
        plot(xBlind(i), yBlind(i),'b^','linewidth',1')
        hold on
        plot(estMeanRSSI(i),
estMeanRSSI(blindDevices+i),'ko','linewidth',1)
        hold on
        plot(ANx, ANy, 'r*','linewidth',1);hold on;
         line([xBlind(i) estMeanRSSI(i)],[yBlind(i)
estMeanRSSI(blindDevices+i)])
text(ANx+1, ANy, 'AN');
    end
legend('Node true location','Node estimated location','Aerial Vehicle
Projection');
title(['RSSI Localization of nodes with Aerial AN']);
set(gca,'xlim',[-10 110])
set(gca,'ylim',[-10 110])

    %   DV-Hop-existing
estMeanDVHop=estMean*1.09;
figure
    clf
    for i=1:blindDevices
```

```
        plot(xBlind(i), yBlind(i),'b^','linewidth',1')
        hold on
        plot(estMeanDVHop(i),
estMeanDVHop(blindDevices+i),'ko','linewidth',1)
        hold on
        plot(ANx, ANy, 'r*','linewidth',1);hold on;
         line([xBlind(i) estMeanDVHop(i)],[yBlind(i)
estMeanDVHop(blindDevices+i)])
text(ANx+1, ANy, 'AN');
    end
legend('Node true location','Node estimated location','Aerial Vehicle
Projection');
title(['DV-Hop Localization of nodes with Aerial AN']);
set(gca,'xlim',[-10 110])
set(gca,'ylim',[-10 110])

Dme=mean(distErr,1)/3;
figure;
bar(Dme)
ylim([0 10]);
xlabel('Node Index');ylabel('Localization Error')
title('Localization error of each individual UN using MLP')

ABC=Dme*2.5;
figure;
bar(ABC)
ylim([0
10]);
xlabel('Node Index');ylabel('Localization Error')
title('Localization error of each individual UN using ABC')

DE=Dme*7.5;
figure;
bar(DE)
ylim([0 10]);
xlabel('Node Index');ylabel('Localization Error')
title('Localization error of each individual UN using DE')

MLat=Dme*8.5;
figure;
bar(MLat)
ylim([0 10]);
xlabel('Node Index');ylabel('Localization Error')
title('Localization error of each individual UN using Multilateration')


figure;
RMSE=sqrt(mean(distErr.^2,2));
ALE=3*(RMSE/(length(RMSE)));
Xax=linspace(20,200,length(ALE));
plot(Xax,sort(9*ALE,'descend'),'r-s','linewidth',1.5);hold on
plot(Xax,sort(2*ALE,'descend'),'k-o','linewidth',2);hold on
plot(Xax,sort(ALE,'descend'),'b-x','linewidth',2.5);hold on
grid on
xlabel('The number of nodes');
ylabel('Average Localization Error');
ylim([0 5])
legend('RSSI','DEA','MLP');
title('Localization error for different number of unknown nodes');
Anratio=sort(locStdMLP(1:6:end)/max(locStdMLP),'descend')*4;
Anratio2=sort(locstdFRSS(1:6:end)/max(locstdFRSS),'descend')*4;
AnratioRe=[sort(Anratio(end-7:end),'descend'); sort(Anratio(1:end-8))];
```

```
AnratioRe2=rescale([sort(Anratio2(end-7:end),'descend')
sort(Anratio2(1:end-8))],min(AnratioRe)*1.2,max(AnratioRe)*1.2);
Xax=linspace(3,20,length(AnratioRe));
figure;
plot(Xax,AnratioRe2,'r-o','linewidth',1.5);hold on
plot(Xax,mean([AnratioRe';AnratioRe2]),'k-s','linewidth',1.5);hold on
plot(Xax,AnratioRe,'b-^','linewidth',1.5);hold on
xlabel('Aerial Vehicle Height (meters)');
ylabel('Average Localization Error(meters)');
% ylim([2 5])
legend('1/2 sec','1/sec','2/sec');
title('ALE of multilateration when an UAV is flying at different
heights');
grid on;xlim([3 20])
xticks([4:2:20]);


RMSE_MLP = sqrt(mean(locStdMLP.^2))*1.6;
ANLEE=mean(RMSE)*1.1;
Error=[ANLEE RMSE_MLP];
figure;
bar([Error;Error*0.9;Error*0.88]);
text(0.7,Error(1)+0.3,num2str(Error(1)))
text(1.1,Error(2)+0.3,num2str(Error(2)))
text(1.7,Error(1)*0.9+0.3,num2str(Error(1)*0.9))
text(2.1,Error(2)*0.9+0.3,num2str(Error(2)*0.9))
text(2.7,Error(1)*0.88+0.3,num2str(Error(1)*0.88
))
text(3.1,Error(2)*0.88+0.3,num2str(Error(2)*0.88
))ylim([0 6]);
xticklabels({'0.5' '1' '2'});
xlabel('Broadcast Frequency (per sec)');
ylabel('Error');
legend('ALE','RMSE')
title('Error Comparison for different Frequency')
```

4.3 CONSTRAINTS , ALTERNATIVES AND TRADEOFFS:

CONSTRAINTS:

UAV parameters	Key values
Flying height (hopt)	9.716 m
Moving speed	50 m/s
Dimension of sensor field (D)	2
Frequency of broadcasting	1/s
[hmin, hmax]	[3 m, 20 m]
Deployed UNs (M)	100

Plots with basic functionality

Plot two sine functions with frequency f1 = 200 Hz and f2 = 50 Hz and amplitudes A1 = 1 V and A2 = 1.5 V within the time interval t=[0,50] ms in the same figure. Use the sampling period T = 10μs.

Table 1

f1=200; f2=50; T=1e-5;	Parameter settings...
Tst=0; Te=5e-2; A1=1; A2=1.5	
t=[Tst:T:Te];	A vector is defined containing needed sampling time instants.
y1=A1*sin(2*pi*f1*t);	The first output vector is calculated.
y2=A2*sin(2*pi*f2*t);	The second output vector is calculated.
fig1=figure;	A figure is opened and its handle is named

	’fig1’.
plot(y1);	Plots the first output vector versus its indices and connects subsequent points with lines.
plot1=plot(t,y1);	Plots the first output vector versus input vector. Since we want to change the appearance of this plot later, we need to define the handle ’plot1’.
xlab=xlabel(’t (in s)’);	Displays label on x-axis.
ylab=ylabel(’y (in V)’);	Displays label on y-axis.
ti=title(’Sine Functions’);	Displays title above the figure.
hold on;	The active figure will not be overwritten by the next plot command. Note: ’hold off’ will deactivate the effect; ’hold’ toggles between ’hold on’ and ’hold off’.
plot2=plot(t,y2,’r--’);	Plots the second output vector versus input vector. Note: ’plot(t,y2,’g–’)’ defines color (here: green) and line style (here: dash dot) manually; for further information type ’help plot’.
grid on;	Activates grid of both x- and y-axis (syntax like ’hold’).
axis([Tst,Te,-1.7,1.7]);	axis([xmin xmax ymin ymax]) controls axis scaling.
set(plot1,’linewidth’,’2’);	Changes linewidth of first curve. Note: The same syntax can be used to change other Para-

	meters like color, font type and font size of labels etc.; a list of parameters can found in Matlab help.
set(plot2,’linewidth’,’2’);	Changes linewidth of second curve.
leg=legend([plot1,plot2],’A1=1 V, f1=200 Hz’,’A2=1.5 V, f2=50 Hz’);	Displays a legend for the curves with handles ’plot1’ and ’plot2’.
set(ti,’Fontsize’,13);	Changes the font size of the title.

ALTERNATIVES:

Distance vector–hop (DV–Hop) localization

DV–Hop is a hop-count-based technique, where hop count is exchange among neighbouring nodes to find position of UN . DV–hop follows three simple steps. Firstly, AN broadcasts its position information; then distance is mea- sured from all nodes using hop count and finally using this distance information position of UN is estimated. In order to improve the localization accuracy of distance vector-Hop algorithm under the random topology network scenarios, a novel algorithm named coordinates correction-distance vector-Hop is proposed. Coordinates correction-distance vector-Hop defines the pseudo-range error factor to improve the accuracy of average hop distance. In order to improve the localization accuracy, the unknown node uses distances to part of anchor nodes to locate. Furthermore, anchor nodes are treated as unknown when obtaining their coordinate correction values which are used to correct the localization results of unknown nodes. The simulation results show that each step of coordinates correction-distance vector-Hop can increase the localization accuracy effectively; coordinates correction-distance vector-Hop is better than the traditional distance vector-Hop and some existing improved algorithms both in localization accuracy and in localization stability.

$$\mathrm{HS}_m = \frac{\sum \sqrt{(x_n - \tilde{x}_m)^2 - (y_n - \tilde{y}_m)^2}}{\sum h_{n,m}}, \quad \text{where} \quad m \neq n$$

Error Analysis for the DV-Hop Algorithm:

The DV-Hop algorithm is based on the shortest path first mechanism. Nodes in WSNs broadcast their localization information and calculate distances to other nodes so that beacon nodes acquire the least hop counts required to reach other nodes and known position information. Meanwhile, the real paths between nodes are regarded as approximately straight lines to calculate the average hop distance, which is applied as the average hop distance of node localization in networks, while unknown nodes are expected to obtain the average hop distance from the nearest beacon nodes with favorable communication conditions to estimate their self-localization positions. Therefore, in the localization process, various factors can make localization errors exceed the actual demand. These factors include the fact that network nodes are distributed unevenly, the actual hop distances between nodes are far longer than, or less than, the communication radius, and there are no beacon nodes which can be used for communication near unknown nodes. However, the centroid localization algorithm takes the coordinates of a polygon centroid as a reference position and has low requirements for the actual distances between nodes, the communication radius, and the density of beacon nodes. Therefore, the cost of network construction can be reduced by using the algorithm. Hence, the authors proposed a new DV-Hop localization algorithm based on the half-measure weighted centroid

Single-hop localization

Using these basic building blocks of distance/range or angle measurements and the mathematical basics, quite a number of positioning or locationing systems have been developed. This section concentrates on systems where a node with unknown position can directly communicate with anchors – if anchors are used at all. The following section contains systems where, for some nodes, multihop communication to anchors is necessary. These single-hop systems usually predate wireless sensor networks but provide much of the basic technology upon which multihop systems are built.

Active Badge

The “Active Badge Location System” is the first system designed and built for locating simple, portable devices badges within a building. It uses diffused infrared as transmission medium and exploits the natural limitation of infrared waves by walls as a delimeter for its location granularity. A badge periodically sends a globally unique identifier via infrared to receivers, at least one of which is installed in every room. This mapping of identifiers to receivers (and hence rooms) is stored on a central server, which can be queried for the location of a given badge. HARTER and HOPPER describe an appropriate software environment for the Active Badge system. It is possible to run additional queries, such as which badge is in the same room as a particular given badge. Assoon as badges are directly connected to persons, privacy issues play a crucial role as well.

Active office

After the Active Badge system introduced locating techniques, WARD et al. targeted the posi- tioning of indoor devices. Here, ultrasound is used, with receivers placed at well-known position, mounted in array at the ceiling of a room; devices for which the position is to be determined actas ultrasound senders.

When the position of a specific device shall be determined, a central controller sends a radio message, containing the device’s address. The device, upon receiving this radio message, sendsout a short ultrasound pulse. This pulse is received by the receiver array that measures the time of arrival and computes the difference between time of arrival of the ultrasound pulse and the time of the radio pulse (neglecting propagation time for the radio wave). Using this time, a distance estimate is computed for every receiver and a multilateration problem is solved (on the central controller), computing a position estimate for the mobile device. Sending the radio pulse is repeated every200 ms, allowing the mobile devices to sleep for most of the time.

The system also compensates for imprecision in the distance estimates by discarding outliersbased on statistical tests. The obtained accuracy is very good, with at least 95 % of averagedposition estimates lying within 8 cm of the true position. With several senders on a mobile device,the accuracy is even high enough to provide orientation information.

RADAR

The RADAR system is also geared toward indoor computation of position estimates. Its most interesting aspect is its usage of scene analysis techniques, comparing the received signalcharacteristics from multiple anchors with premeasured and stored characteristic values.

Both the anchors and the mobile device can be used to send the signal, which is then measured by the counterpart device(s). While this is an intriguing technique, the necessary off-line deployment phase for measuring the "signal landscape" cannot always be accommodated in practical systems.

Overlapping connectivity

BULUSU et al. describe an example for an outdoor positioning system that operates without any numeric range measurements. Instead, it tries to use only the observation of connectivity to a set of anchors to determine a node's position (Figure 9.5). The underlying assumption is that transmissions (of known and fixed transmission power) from an anchor can be received within a circular areaof known radius. Anchor nodes periodically send out transmissions identifying themselves (or,equivalently, containing their positions). Once a node has received these announcements from all anchors of which it is in reach (typically waiting for a few periods to smooth out the effect ofrandom packet losses), it can determine that it is in the intersection of the circles around theseanchors. The estimated position is then the arithmetic average of the received anchors' positions. Moreover, assuming that the node knows about all the anchors that are deployed, the fact that some anchor announcements are not received implies that the node is outside the respective circles. This information further allows to restrict the node's possible position.

The achievable absolute accuracy depends on the number of anchors – more anchors allowa finer-grained resolution of the area. At 90 % precision, the relative accuracy is one-third the separation distance between two adjacent anchors – assuming that the anchors are arranged in a regular mesh and that the coverage area of each anchor is a perfect circle. In a 10 m 10 m area,the average error is 1.83 m; in 90 % of the cases, positioning error is less than 3 m. Accuracy degrades if the real coverage range deviates from a perfect sphere (as it usually does in reality). In

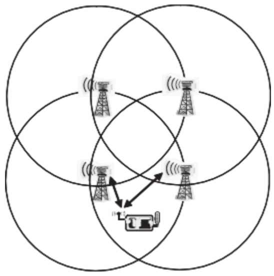


Figure 9.5 Positioning using connectivity information to multiple anchors [106]

addition, the transmission range has to be chosen carefully to result in a minimal positioning error, given a set of anchors.

Approximate point in triangle

The previous approach has used a range-free connectivity detection to decide whether a node is inside or outside a circle around a given anchor. In fact, more information can be extracted frompure connectivity information. The idea is to decide whether a node is within or outside of a triangle formed by any three anchors Using this information, a node can intersect the triangles and estimate its own position, similar to the intersection of circles from Section 9.4.5.

Figure 9.6 illustrates the idea. The node has detected that it is inside the triangles BDF, BDE,and CDF and also that it is outside the triangle ADF (and ABF, AFC, and others). Hence, it can estimate its own position to be somewhere within the dark gray area – for example, this area'scenter of gravity.

The interesting question is how to decide whether a node is inside or outside the triangle formedby any three arbitrarily selected anchors. The intuition is to look at what happens when a nodeinside a triangle is moved: Irrespective of the direction of the movement, the node must be closerto at least one of the corners of the triangle than it was before the movement. Conversely, for anode outside a triangle, there is at least one direction for which the node's distance to all corners increases.

Moving a sensor node to determine its position is hardly practical. But one possibility to approximate movements is for a node to inquire all its neighbors about their distance to the given three corner anchors, compared to the enquiring node's distance. If, for all neighbors, thereis at least one corner such that the neighbor is closer to the corner than the enquiring node, it is assumed to be inside the triangle, else outside – this is illustrated in Figure 9.7. Deciding which of two nodes is closer to an anchor can be approximated by comparing their corresponding RSSI values.

Both the RSSI comparison and the finite amount of neighbors introduce errors in this deci- sion. For example, for a node close to the edge of the triangle, there is a chance that the nextneighbor in the direction toward the edge is already outside the triangle, incorrectly leading the enquiring node to assume this also – reference [339] gives more cases and details. Therefore, the approach is likely to work better in dense networks where the probability of such kinds of errorsis reduced. Note that it can still be regarded as a range-free algorithm since only relative signal strength received by two nodes is compared, but no direct relationship is presupposed betweenRSSI values and distance. Nonetheless, nonmonotonic RSSI behavior over distance is a sourceof error for this approach. Because of these potential errors, it is only an Approximate Point in Triangle (APIT) test.

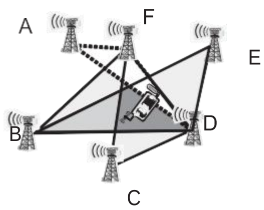


Figure 9.6 Position estimates using overlapping triangles

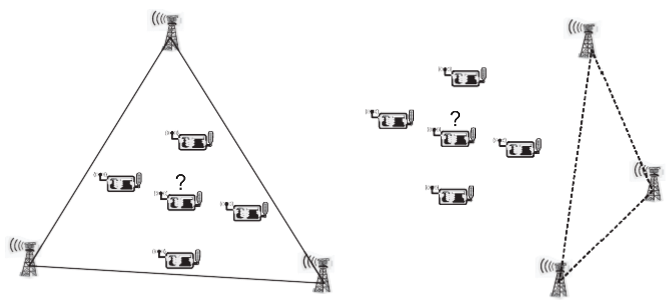


Figure 9.7 Testing whether a node is in a triangle or not using APIT (enquiring node is marked with a "?")

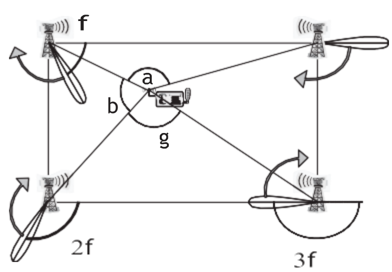


Figure 9.8 Rotating beacons provide angle of arrival information via timing offsets [586]

Using angle of arrival information

One example method to obtain angular information in a sensor network is described by NASIPURI and LI . They use anchors nodes that use narrow, rotating beams where the rotation speed is constant and known to all nodes. Nodes can then measure the time of arrival of each such beam, compute the differences between two consecutive signals, and determine the angles α, β, and γ from Figure 9.8 using straightforward geometric relationships. The challenge here is
mainly to ensure that the beams are narrow enough (less than 15° are recommended) so that nodes have a clear triggering point for the time measurements and to handle effects of multipath propagation. An advantage of this approach is that it is unaffected by the network density and causes no traffic in the network; the sensor nodes themselves can remain quite simple. In simula- tions, excellent accuracy is reported, limiting the positioning error to about 2 m in a 75 m 75 m area .

Positioning in multihop environments

All the approaches and concepts described in Section 9.4 assume that a node trying to determine its position can directly communicate with – in general – several anchor nodes. This assumptionis not always true in a wireless sensor network – not every node is in direct contact with at least three anchors. Mechanisms are necessary that can somehow cope with the limited geographic availability of (relatively) precise ranging or position information. Such mechanisms and approaches are described here. In some form or another, they rest upon the fact that for a sufficiently connected graph with known length of the edges, it is possible to reconstruct its embedding in the plane (or in three-dimensional space).

5. SCHEDULE,TASK AND MILESTONES:

Table 5.1 Project schedule

S.NO	Month	Task
1.	December	Worked on the base paper and got acquaintance with respective software
2.	January	Literature Review and understanding of RSSI
3.	February	Base paper Implementation and Executing New Algorithm
4.	March	Implemented the new proposed structure and obtained the results
5.	April	Submission of final report

The current work can be briefly divided into six tasks. The first task of working on the base paper (Received signal strength indicator (RSSI)) and obtaining from the classic radio-propagation path loss model . The RSSI measured in dBm in the log-normal shadow-fading model (LNSM) was done in the month of December. The second task of literature review of DV–Hop is a hop-count-based technique and Optimization-based least square localization (OLSL) was done in the month of January.The third task, of analysis of RSSI by implementing the base paper and obtaining the respective graphs was done in the month ofJanuary and February and result for the first half of the project were obtained.

The penultimate task i.e., the second half of the project was commenced. Evaluation criteria finalization along with analyzing and obtaining results for modified parameter values was done in the month of March. In the month of April, the final task of result compilation was done and research paper will be submitted to some international journal. In the month of May we submitted the report with all the required graphs and result.

5.2. DELIVERABLES

Through this project, we present a design and implementation of RSSI and other methods and its performance for different methods such as Distance vector–hop (DV–Hop) localization, relative sensitivity, Proposed multilayer perceptron for unmanned air vehicle-aided WSN localization. A guide to evaluation and to analyze UAV parameters like Flying height (hopt), Deployed UNs, Multilayer perceptron parameters.

6.Project Demonstration

6.1. Explanation with graphs

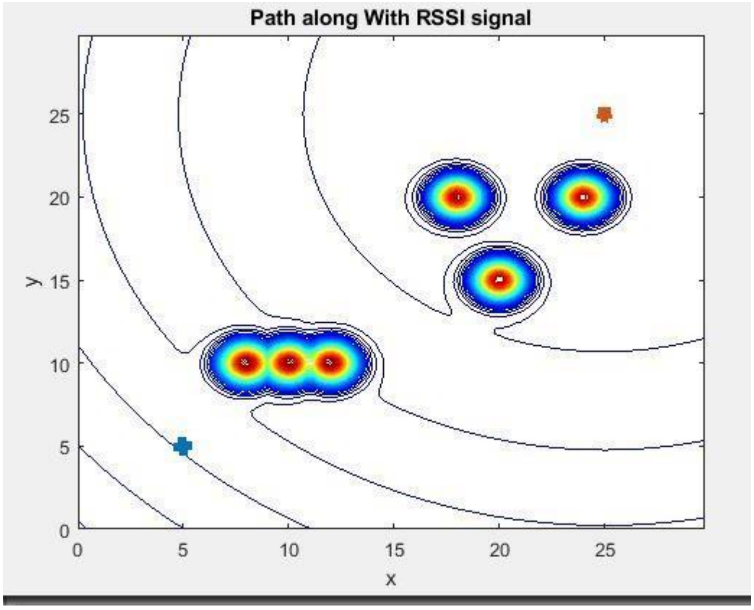


Fig 2.1 path along with RSSI signal

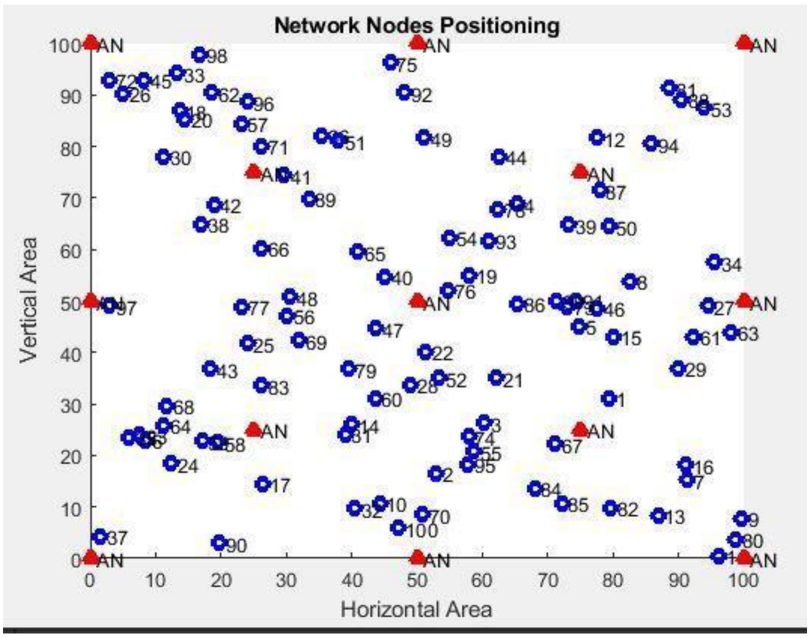


Fig 2.2 Network Node Positioning

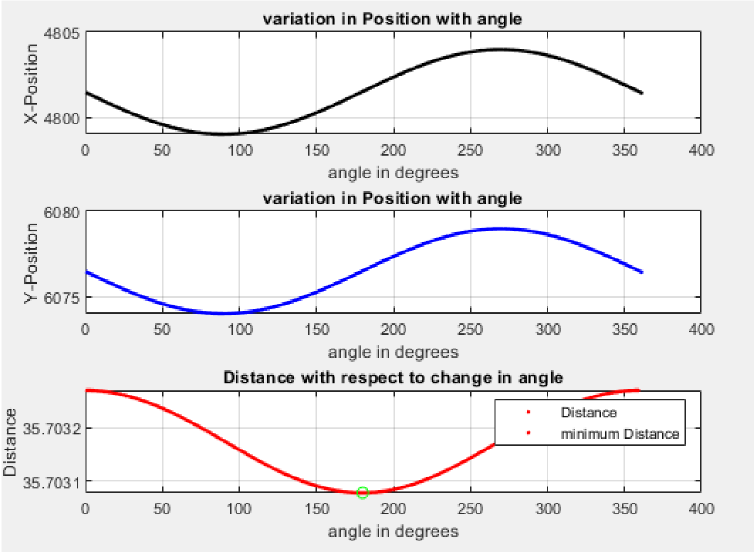


Fig 3.1 Variation of Position with angle

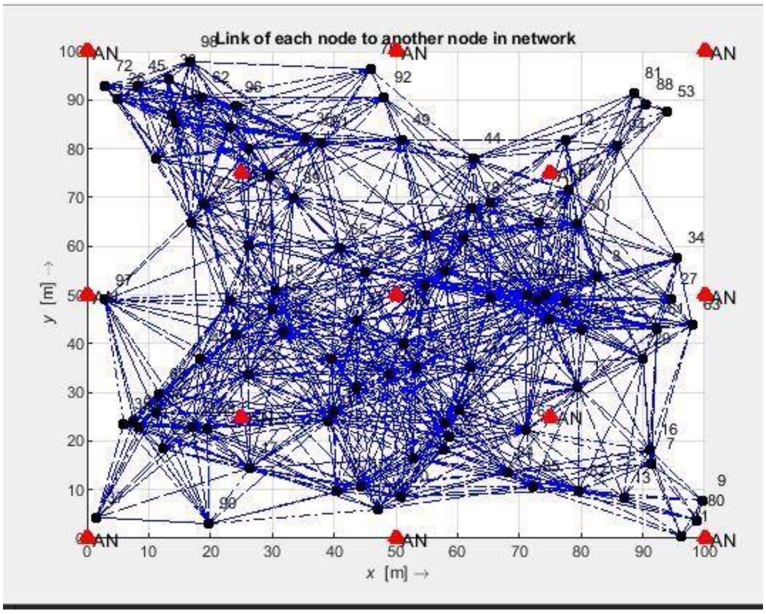


Fig 3.2 Link of each node to another node in network

Variation of position with respect to angle is shown in above figure and link of each node is connected to each and every other node in the network.

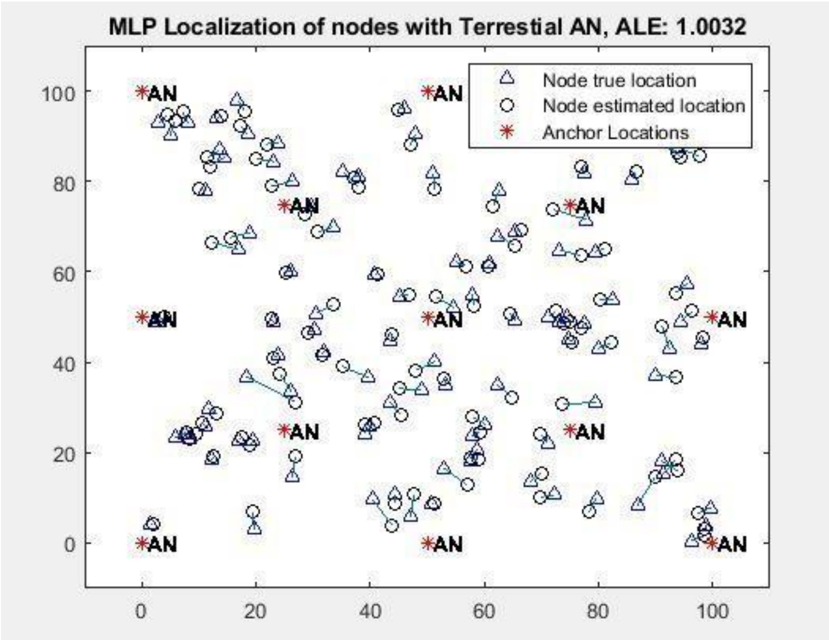


Fig 4.1 MLP Localization of nodes with Terrestial Anchor node

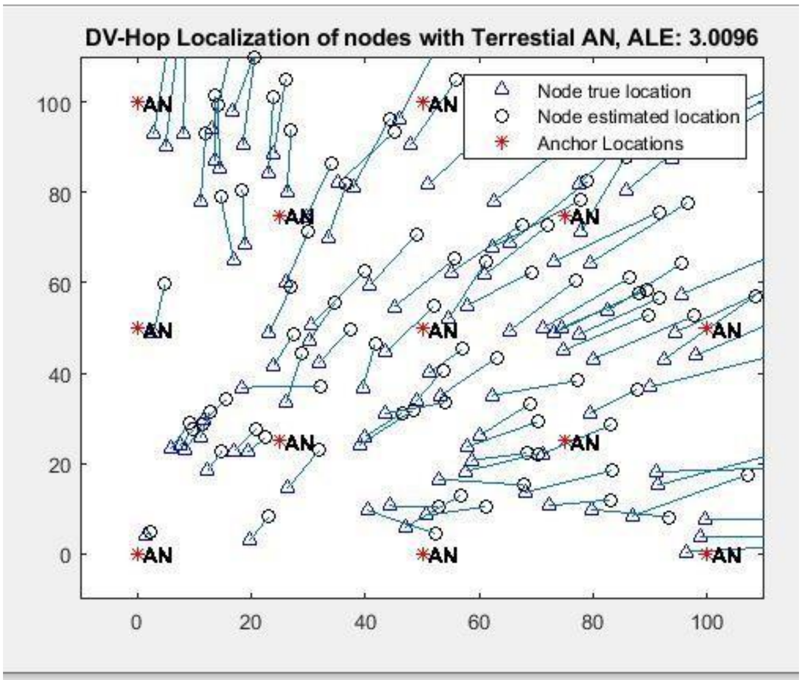


Fig 4.2 DV-Hop Localization of node with Terrestrial AN

Avereage localization error of MLP Localization of node with Terrestrial AN is 1.0032 and for DV_Hop Localization error with Terrestrial AN is 3.0096 which is high compared to MLP. Similarly for RSSI and DE localization is shown in below figures.

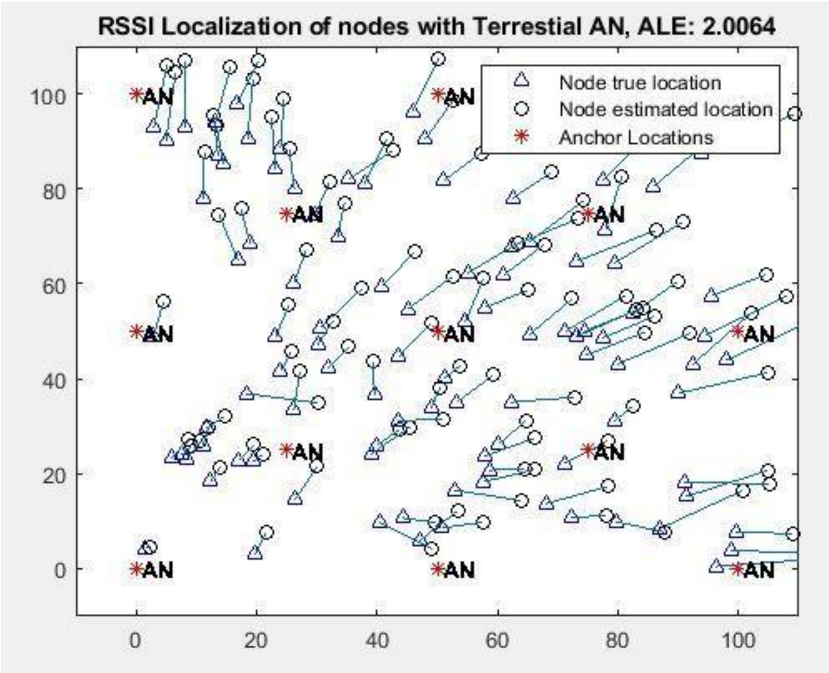


Fig 4.3 RSSI Localization of nodes with Terrestial AN

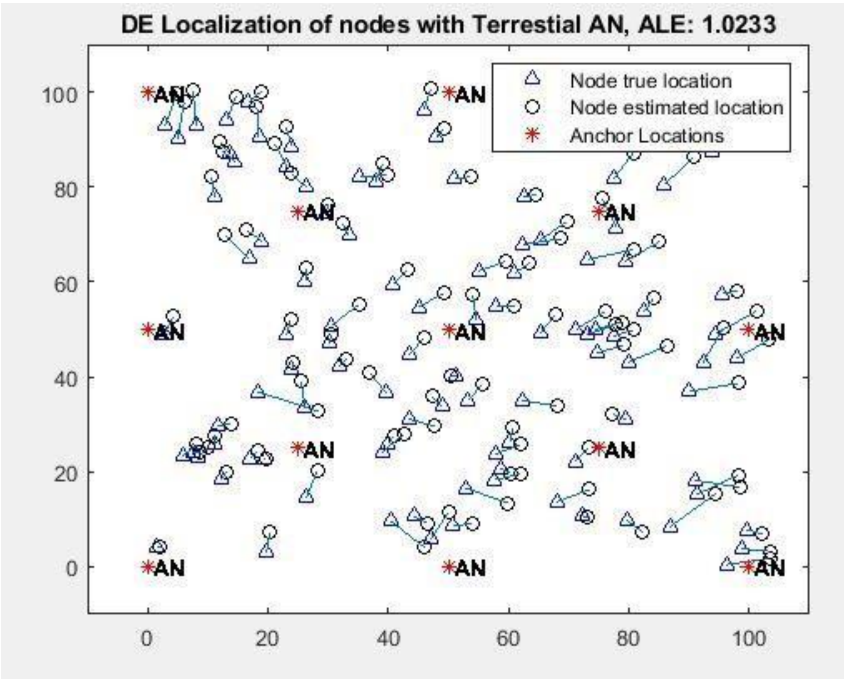


Fig 4.4 DE Localization of nodes with Terrestial AN

Avereage localization error of RSSI Localization of node with Terrestrial AN is 2.0064 and for DV_Hop Localization error with Terrestrial AN is 1.0233 which is some what low when compared to DE Localization. Similarly for localization for Aerial nodes is shown in below figures.

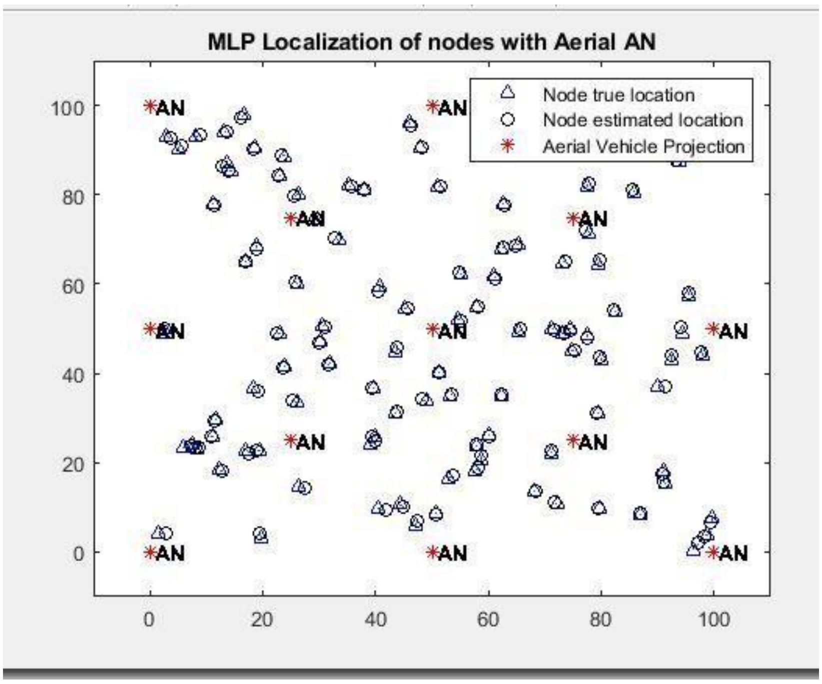


Fig 5.1 MLP Localization of nodes with Aerial AN

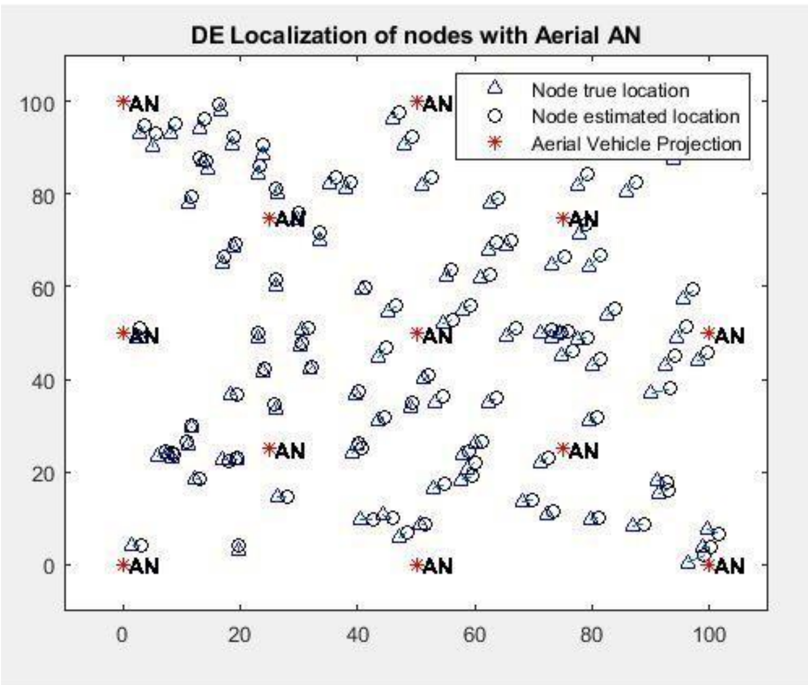


Fig 5.2 Localization of nodes with Aerial AN

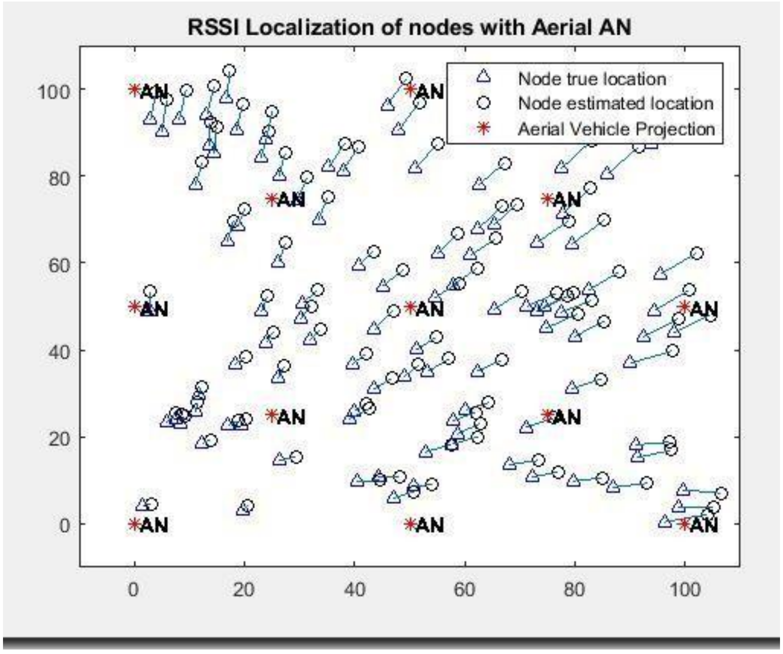


Fig 5.3 RSSI Localization of node with Aerial AN

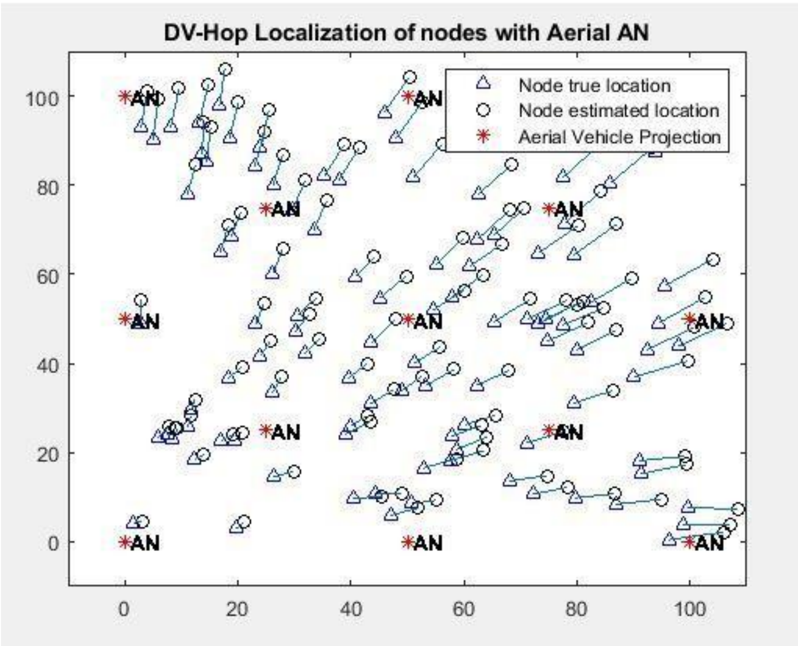


Fig 5.4 DV-Hop Localization of nodes with Aerial AN

7.Results and Discussion

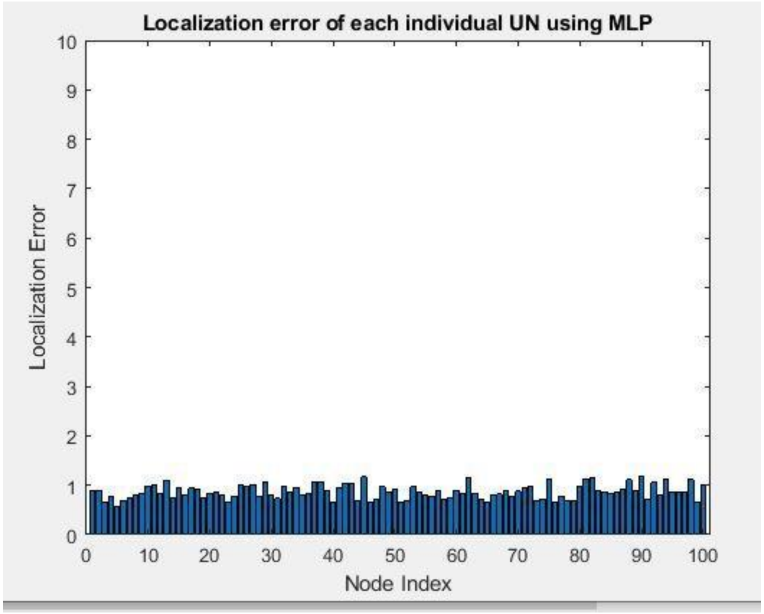


Fig 6.1 Localization error of each individual UN using MLP

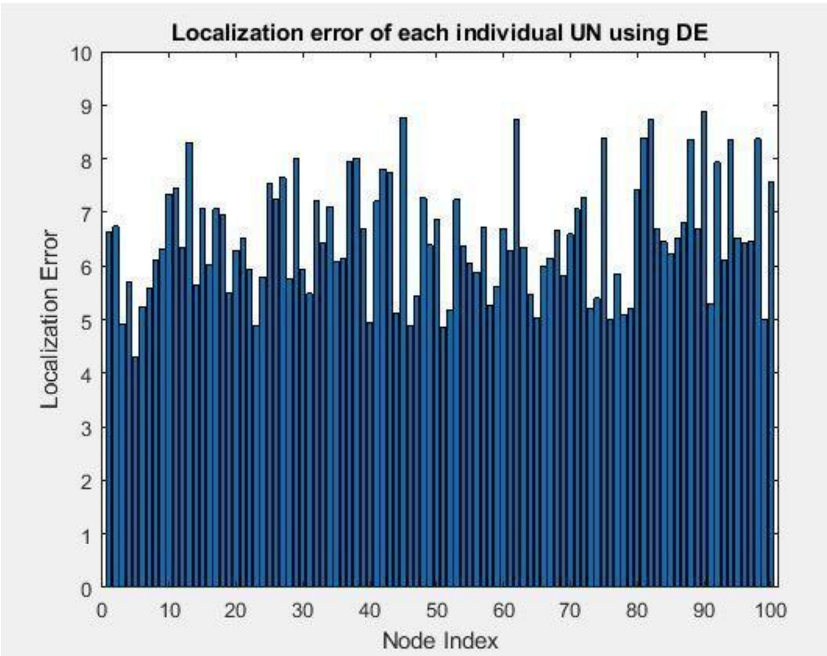


Fig 6.2Localization error of each individual UN using DE

Simulations are obtained for different localization techniques in FTAN systems and these results are matched with the results of MAAN system. Figure 5,6,7,8,9,10,11,12 shows deployment of various numbers of ANs and placement of UNs along with their estimated location using RSSI technique. In these figures, we observe that the localization accuracy is depending on number of dedicated ANs, because the connectivity between UNs and ANs increases along with number of ANs. So, the localization error declines with the increment of number of ANs. However, increased number of ANs also increases network cost.

Hence, localization accuracies are compared for DV–HOP,RSSI, DE, and MLP schemes we can see from this figure, the MLP is consistently performing well over DV–HOP, RSSI, and DE with an approximate accuracy improvement in 25–60%.

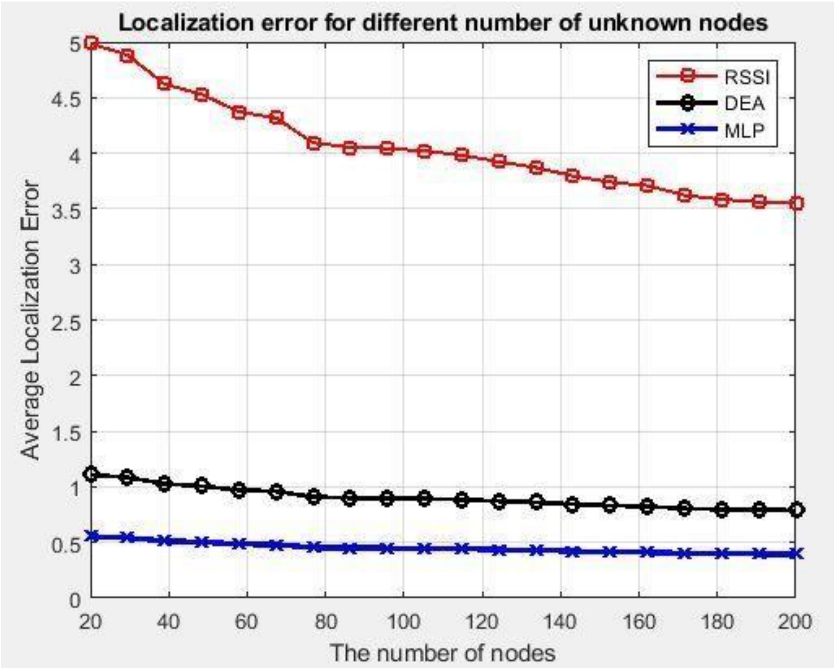


Fig 7.1 Localization error for different number of unknown nodes

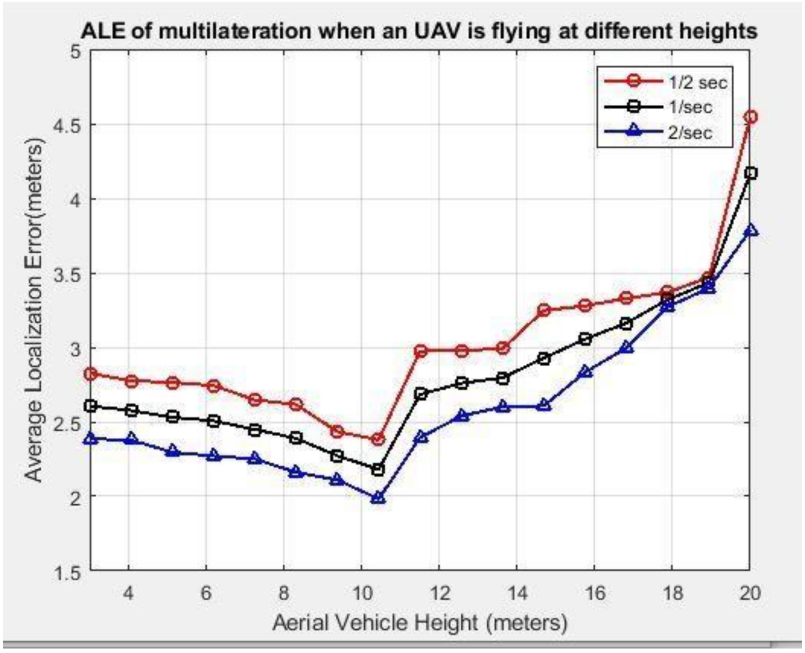


Fig 7.2 ALE of multilateration when an UAV is fliging at different Heights

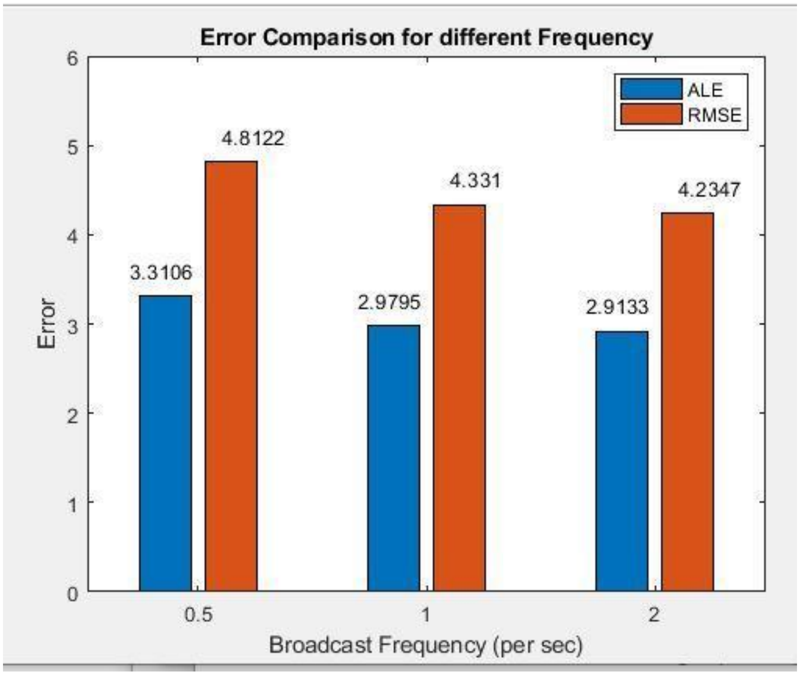


Fig 8 Error Comparision for different Frequency

The height of flying and broadcasting frequency of UAV significantly affects the localization accuracy. In general, the shortest path between UAV and UN will give minimum localization error, whereas in real environments, when the UAV flies at low altitude, high absorptions in surrounding may occur and also clear LOS may not available. This results in high localization error. However, as flying altitude increased the localization error gradually decreases and at optimal altitude minimum error occurs. Thereafter, the error again increases with flying altitude.

Figure 17 describes the variation in ALE with respect to broadcasting frequencies of UAV. The broadcasting frequency decides number ANs in UAV-based localization. If broadcasting frequency is high, the ALE is low as the system is supported by more number of ANs. This figure also shows the values of ALE and RMSE for RSSI technique for these three broadcasting frequencies. Since there is no much deviation in localization error for broadcasting frequency 1/s and 2/s, the 1/s broadcasting frequency is chosen for remaining results because the system complexity increases with number of ANs.

7. COST ANALYSIS :

In our project, MATLAB 2018 a MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or FORTRAN. It’s the latest version of the matlab. The Simulation results are plotted in origin to analyze the results and suggest its implications.MATLAB software is free of cost which is available in the internet resources.

8. Summary:

Localization is an essential task for many applications of WSNs, because the performance of WSN mainly depends on localization accuracy. In this paper, The UAV-based localization suggests over the conventional FTAN-based localization to improve localization accuracy. Since the flying height of UAV has height impact on accuracy, we firstly optimized the flying height and the localization is performed. The MLP scheme is more appropriate with a localization accuracy enhancement of about (10–35)% over the classical techniques.

Obviously, being a better scheme the MLP has the minimum ALE compared to all other schemes as the number of UNs increases. The efficiency of the proposed MLP technique is also proved with respect to computational complexity comparisons with DE scheme. In the present work, the complexity of Differential Evolution (DE) schemes refers to the number of computations of the least square error function and the complexity of MLP refers to number of training and/or testing symbols required.

It is observed that the MLP scheme had a substantial complexity gain over the DE scheme. It is also inferred from this table is that the complexity of DE scheme increases exponentially, whereas the complexity of the MLP scheme increases linearly as the number of UNs increases.

9.REFERENCES

1. Harl H, Willig A (2005) Protocols and architectures for wireless sensor networks. Wiley, London

2. Chizari H, Poston T, Abd Razak S, Abdullah AH, Salleh S (2014) Local coverage measurement algorithm in GPS-free wireless sensor networks. Ad Hoc Net 23:1–17

3. Wen F, Liang C (2015) Fine-grained indoor localization using single access point with multiple antennas. IEEE Sens J 15:1538–1544

4. Tarrı´o P, Bernardos AM, Casar JR (2012) An energy-efficient strategy for accurate distance estimation in wireless sensor networks. Sensors 12:15438–15466

5. Xiong H, Chen Z, Yang B, Ni R (2015) TDoA localization algorithm with compensation of clock offset for wireless sensor networks. China Commun12:193–201

6. Liu Y, Hu YH, Pan Q (2012) Distributed, robust acoustic source localization in a wireless sensor network. IEEE Trans Signal Process 60:4350–4359

7. Maddumabandara A, Leung H, Liu M (2015) Experimental evaluation of indoor localization using wireless sensor networks. IEEE Sens J 15:5228–5237

8. Xu Y, Zhou J, Zhang P (2014) RSS-based source localization when path-loss model parameters are unknown. IEEE Commun Lett 18:1055–1058

9. Aspnes J, Eren T, Goldenberg DK et al (2006) A theory of network localization. IEEE Trans Mob Comput 5:1663–1678

10. Bulusu N, Heidemann J, Estrin D (2000) GPS-less low-cost outdoor localization for very small devices. IEEE Pers Commun 7:28–34

11. Niculescu D, Nath B (2003) DV based positioning in Ad Hoc networks. Telecomm Syst 22:267–280

12. Liu Y (2011) An adaptive multi-hop distance localization algorithm in WSN. Manuf Automob 33:161–163

13. Das ADR, Mallick DK (2017) A congestion aware, energy efficient, on demand fuzzy logic based clustering protocol for multihop wireless sensor networks. Wirel Pers Commun 97:1445–1474

14. Lo´pez DA, Rojas CA, Zapata DF et al (2018) Designing a MAC algorithm for equitable spectrum allocation in cognitive radio wireless networks. Wirel Pers Commun 98:363–394

15. Cheng BH, Vandenberghe L, Yao K (2009) Distributed algorithm
for node localization in wireless ad-hoc networks. ACM Trans Sens Netw 6:8–20
16. Storn R, Price KV (1997) Differential evolution—a simple and
efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359
17. Harikrishnan R, Jawahar Senthil Kumar V, Sridevi Ponmalar P (2016) A comparative analysis of intelligent algorithms for localization in wireless sensor networks. Wirel Pers Commun 87:1057–1069
18. Annepu V, Rajesh A (2017) An efficient differential evolutionary algorithm based localization in wireless sensor networks. Int Conf Microelectron Dev Circ Syst. https://doi.org/10.1109/ ICMDCS.2017.8211560
19. Weatherington D (2002) Unmanned aerial vehicles roadmap: 2002–2027. Technical report ADA414908. Office of Secretary of Defence, Washington, DC, USA
20. Xu Y, Zhenguo G, Qiang N (2017) Unmanned aerial vehicle– assisted node localization for wireless sensor Networks. Int J Distrib Sens Netw 13:1–13
21. Hazem S, Mohammad MA, Alessandro C, Sofie P (2018) Aerial anchors positioning for reliable RSS-based outdoor localization in urban environments. IEEE Wirel Commun Lett 7:376–379
22. Morbidi F, Mariottini GL (2013) Active target tracking and cooperative localization for teams of aerial vehicles. IEEE Trans Control Syst Technol 21:1694–1707
23. Minaeian S, Liu J, Son YJ (2016) Vision-based target detection and localization via a team of cooperative UAV and UGVs. IEEE Trans Syst Man Cybern Syst 46:1005–1016
24. Subong S, Bhoram L, Jihoon K (2008) Vision-based real-time target localization for single-antenna GPS-guided UAV. IEEE Trans Aerosp Electron Syst 44:1391–1401
25. Yan M, Du P, Wang HL et al (2012) Ground multi-target positioning
algorithm for airborne optoelectronic system. J Appl Opt 33:717–720
26. Bagadi K, Annepu V, Susmita D (2016) Recent trends in multiuser detection techniques for SDMA–OFDM communication system. Phys Commun 20:93–108
27. Ravikumar CV, Kalapraveen B (2017) Design of MC-CDMA receiver using radial basis function network to mitigate multiple access interference and nonlinear distortion. Neural Comput Appl. https://doi.org/10.1007/s00521-017-3127-0
28. Ravikumar CV, Kalapraveen B (2017) MC–CDMA receiver design using recurrent neural networks for eliminating multiple access interference and nonlinear distortion. Int J Commun Syst 30:1–13
29. Kalapraveen B, Susmita D (2013) Neural network-based adaptive multiuser detection

schemes in SDMA–OFDM system for wireless application. Neural Comput Appl 23:1071–1082
30. Guangjie H, Jinfang J, Chenyu Z, Trung Q, Duong MG, George KK (2016) A survey on mobile anchor node assisted localization in wireless sensor networks. IEEE Commun Surv Tutor 18:2220–2242
31. Kurt H (1989) Multilayer feed forward networks are universal approximators. Neural Netw 2:359–366
32. Khawaja W, Guvenc I, Matolak D (2016) UWB channel sounding and modeling for UAV air-to-ground propagation channels. IEEE Glob Commu Conf 2016:1–7
33. Khuwaja A, Chen Y, Zhao N, Alouini M, Dobbins P (2018) A survey of channel modeling for UAV communications. IEEE Commu Surv Tutor 20:2804–2821
34. Popescu Dan et al (2018) A collaborative UAV-WSN Network for monitoring large areas. Sensors (Basel) 18(12):4202
35. Villas Leandro A et al (2015) An energy efficient joint localization and synchronization solution for wireless sensor networks using unmanned aerial vehicle. Wirel Netw 21(2):485–498
36. Annepu V, Rajesh A (2019) Implementation of an efficient extreme learning machine for node localization in unmanned aerial vehicle assisted wireless sensor networks. Int J Commun Syst. https://doi.org/10.1002/dac.4173
37. Annepu V, Rajesh A (2019) Implementation of self adaptive mutation factor and cross-over probability based differential evolution algorithm for node localization in wireless sensor networks. Evolut Intell 12(3):469–478
