เอกสารอ้างอิง

1. Stansfield, R. G. Statistical theory of DF fixing. Journal of the IEE (London) 94(15) (1947): 762-770.
2. Wax, M. Position Location From Sensors with Position Uncertainty.IEEE Transactions on Aerospace. (September 1983) : 658-662.
3. Torrieri, D. J. Statistical theory of Passive Location Systems.IEEE Transactions on Aerospace. (March 1984) : 183-198.
4. Spingarn, K. Passive Position Location Estimation Using the Extended Kalman Filter.IEEE Transactions on Aerospace. (July 1987) : 558-567.
5. Aidala, V. Kalman Filter Behavior in Bearing-Only Tracking Applications.IEEE Transactions on Aerospace. (January 1979) : 29-39.
6. Paul, C. Chestnut. Emitter Location Accuracy Using TDOA and Differential Doppler. IEEE Transactions on Aerospace.(March 1982) : 214-218.
7. Srdjan, S. Stankovic, Marina Dragosevic, and Miodrag Carapic. Estimation of Sinusoids Using The Extended Kalman Filter. IEEE ICASSP. (1984): 5.11.1-5.11.4.
8. Bertrand, T. Fang. Simple Solutions For Hyperbolic and Related Position Fixes.IEEE Transactions on Aerospace. (September 1990) : 748-749.
9 Poirot, J. and McWilliams, G. Navigation by Back Triangulation.IEEE Transactions on Aerospace. (March 1976) : 270-274.
10. Linnik, Y. Method of LS & principles of the theory of observations, pp.10-11. Pergamon Press, N.Y., 1961.
11. Jame, V. Candy. Maximum a Posteriori/MaximumLikelihoodEstimation, pp.61-62. Signal Processing The Model-Based Apporoach, 1987.
12. Robert, Grover, Brown. The Discrete Kalman Filter, pp.195-200. Introduction To Random Signal And Kalman Filtering, 1983
13. Balakrishnan, A. Kalman Filtering Theory, Optimization Software, Inc. New York,1987.
14. Jazwinski, A. Stochastic Process & Filtering, NewYork:Acadimic Press, 1970.
15. Goodwin, G & K. Siy. Adaption Filtering Prediction & Control, Prentice-Hall, New Jersey, 1984.

16. Kalada, R. and Spingarn, K. Control Identification and Input Optimization, New York : Plenum Press, 1982.

17. Rhein, M. Handbook on Weaponry, Gmbh, Dusseldort, 1982

18. David A. Shnidman. Effcient Computation of the Circular Error Probability(CEP)Integral. IEEE Transactions on Automatic Control.(August 1995) : 1472-1474.

ภาคผนวก ก

โปรแกรม การจำลองบนคอมพิวเตอร์

การสร้างอินพุทของระบบ

1. <u>กรณีที่มี 2 observers</u>

```
% DIRECTION FINDING PROJECT
% 2 PASSIVE STATION SIMULATION MODULE
% CREATE BEARING ANGLES

% initialize all variables

% Station location
 source=[15 15];

 station_i=[0 0;0 0];    % stations' location matrix
         a=[0 0 ];       % angle matrix
       row=2;            % define number of row

 for  j=1:row
   a(1,j)=input('angle ');
   station_i(j,1)=input('x position ');
   station_i(j,2)=input('y position ');
 end

% start 1 to 40 observations

  for obsv =1:40      % 40 observations
```

```
% A 100 montecarlo run start point

  reff=1;          % first point for shifting
  for monte=1:100      %   100 monte carlo runs

  N=randn(obsv,2) ;  % Normalized Gaussian Random Variable

% create matrix b,n and theta

   for i=1:obsv
     for j=1:2
       b(i,j)=0;  % matrix for bearing angle
       n(i,j)=0;  % matrix for normalized Gaussian noise
   theta(i,j)=0;  % matrix for real angle
     end
   end

   for i=1:obsv
     theta(i,1)=a(1,1); % copy real angles
     theta(i,2)=a(1,2);
   end

     n=N;        % Gaussian noise
     b=theta+n;  % add noise

% store n into Zi
      z=n;           %  noise

     if obsv ~=1
      z=(sum(z))/obsv;
     end
```

```
% Noise Storage Matrix

   ZI(obsv,1)=z(1,1);          % noise bearing matrix for all
   ZI(obsv,2)=z(1,2);

   if obsv ~= 1
     sZI=sum(ZI)/obsv;
   else
     sZI=ZI;
   end

   Zi(obsv,reff)=sZI(1,1);
   Zi(obsv,reff+1)=sZI(1,2);

   if obsv ~=1
    b=(sum(b))/obsv;
   end

% BEARING ANGLE Storage Matrix
 % store b into Bi
   Bi(obsv,reff)   =b(1,1); % bearing matrix for all
   Bi(obsv,reff+1)=b(1,2); %   next processes.

   reff=reff+2;        % 2steps shift

 end % end a 100 monte carlo runs

end    % end of 40 observations loop

%======================end of bearing random=================
```

2. <u>กรณีที่มี 3 observers</u>

```
% DIRECTION FINDING PROJECT
% 3 PASSIVE STATION SIMULATION MODULE
% CREATE BEARING ANGLES

% initialize all variables

% Station location
 source=[15 15];

 station_i=[0 0;0 0];    % stations' location matrix
         a=[0 0 ];       % angle matrix
     row=3;              % define number of row

 for  j=1:row
   a(1,j)=input('angle ');
   station_i(j,1)=input('x position ');
   station_i(j,2)=input('y position ');
 end

% start 1 to 40 observations

  for obsv =1:40      % 40 observations

% A 100 montecarlo run start point

  reff=1;       % first point for shifting
  for monte=1:100     %   100 monte carlo runs

   N=randn(obsv,3) ;  % Normalized Gaussian Random Variable
```

```
% create matrix b,n and theta

  for i=1:obsv
    for j=1:3
      b(i,j)=0;  % matrix for bearing angle
      n(i,j)=0;  % matrix for normalized Gaussian noise
  theta(i,j)=0;  % matrix for real angle
    end
  end

  for i=1:obsv
    theta(i,1)=a(1,1); % copy real angles
    theta(i,2)=a(1,2);
    theta(i,3)=a(1,3);
  end

    n=N;       % Gaussian noise
    b=theta+n;  % add noise

% store n into Zi
     z=n;            % noise

    if obsv ~=1
     z=(sum(z))/obsv;
    end

 % Noise Storage Matrix
    ZI(obsv,1)=z(1,1);          % noise bearing matrix for all
    ZI(obsv,2)=z(1,2);
    ZI(obsv,3)=z(1,3);
```

```
    if obsv ~= 1
      sZI=sum(ZI)/obsv;
    else
      sZI=ZI;
    end

    Zi(obsv,reff)=sZI(1,1);
    Zi(obsv,reff+1)=sZI(1,2);
    Zi(obsv,reff+2)=sZI(1,3);

    if obsv ~=1
     b=(sum(b))/obsv;
    end

 % BEARING ANGLE Storage Matrix
  % store b into Bi
    Bi(obsv,reff)   =b(1,1); % bearing matrix for all
    Bi(obsv,reff+1)=b(1,2); %   next processes.
    Bi(obsv,reff+2)=b(1,3);

    reff=reff+3;        % 2steps shift

  end % end a 100 montecarlo run

 end    % end of observation loop

%=====================end of bearing random================
```

3. กรณีที่มี 4 observers

```
% DIRECTION FINDING PROJECT
% 4 PASSIVE STATION SIMULATION MODULE
% CREATE BEARING ANGLES

% initialize all variables

% Station location
 source=[15 15];

 station_i=[0 0;0 0];   % stations' location matrix
        a=[0 0 ];       % angle matrix
     row=4;             % define number of row

 for  j=1:row
   a(1,j)=input('angle ');
   station_i(j,1)=input('x position ');
   station_i(j,2)=input('y position ');
 end

% start 1 to 40 observations

 for obsv =1:40     % 40 observations

% A 100 montecarlo run start point

 reff=1;       % first point for shifting
 for monte=1:100    %   100 monte carlo runs

  N=randn(obsv,4) ;  % Normalized Gaussian Random Variable
```

```
% create matrix b,n and theta

   for i=1:obsv
     for j=1:4
       b(i,j)=0;  % matrix for bearing angle
       n(i,j)=0;  % matrix for normalized Gaussian noise
   theta(i,j)=0;  % matrix for real angle
     end
   end

   for i=1:obsv
     theta(i,1)=a(1,1); % copy real angles
     theta(i,2)=a(1,2);
     theta(i,3)=a(1,3);
     theta(i,4)=a(1,4);
   end

     n=N;        % Gaussian noise
     b=theta+n;  % add noise

% store n into Zi
     z=n;          %  noise

    if obsv ~=1
     z=(sum(z))/obsv;
    end

 % Noise Storage Matrix
    ZI(obsv,1)=z(1,1);          % noise bearing matrix for all
    ZI(obsv,2)=z(1,2);
    ZI(obsv,3)=z(1,3);
    ZI(obsv,4)=z(1,4);
```

```
if obsv ~= 1
  sZI=sum(ZI)/obsv;
else
  sZI=ZI;
end

Zi(obsv,reff)=sZI(1,1);
Zi(obsv,reff+1)=sZI(1,2);
Zi(obsv,reff+2)=sZI(1,3);
Zi(obsv,reff+3)=sZI(1,4);

if obsv ~=1
 b=(sum(b))/obsv;
end

% BEARING ANGLE Storage Matrix
 % store b into Bi
   Bi(obsv,reff)   =b(1,1); % bearing matrix for all
   Bi(obsv,reff+1)=b(1,2); %   next processes.
   Bi(obsv,reff+2)=b(1,3);
   Bi(obsv,reff+3)=b(1,4);

   reff=reff+2;        % 2steps shift

 end % end a 100 montecarlo run

end    % end of observation loop

%=====================end of bearing random================
```

ภาคผนวก ข

โปรแกรม การจำลองบนคอมพิวเตอร์

การประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณ โดยวิธี EXTENDED KALMAN FILTER

1. <u>กรณีที่มี 2 observers</u>

```
% Direction Finding Project
% 2 passive station simulation module
% estimator: EXTENDED KALMAN FILTER

% find priori estimate position

% Station Location
  source=[15 15];

 % find EXTENDED KALMAN FILTER

    for obsv =1:40
     for monte =1:2:200

          X00=[x; y];      % x,y initial
           Xe=[0;0];       % x,y at emitter
            I=[1 0;0 1];   % identity matrix

          Xk1=[0;0];                    % location of sensor1
          Xk2=[station_i(2,1);0];       % location of sensor2

   % transfer bearing to new variables
   Zn(1,1)=Zi(obsv,monte)*pi/180;
```

```
Zn(2,1)=Zi(obsv,monte+1)*pi/180;
   % find nonlinear function

      u1=(X00(2,1)-Xk1(2,1))/(X00(1,1)-Xk1(1,1));
      u2=(X00(2,1)-Xk2(2,1))/(X00(1,1)-Xk2(1,1));

      h11=(-u1/(1+u1^2))*(1/(X00(1,1)-Xk1(1,1)));
      h12=(1/(1+u1^2))*(1/(X00(1,1)-Xk1(1,1)));

      h21=(-u2/(1+u2^2))*(1/(X00(1,1)-Xk2(1,1)));
      h22=(1/(1+u2^2))*(1/(X00(1,1)-Xk2(1,1)));

   An(1,1)=h11;
   An(1,2)=h12;

   An(2,1)=h21;
   An(2,2)=h22;

   Wn=inv((An)'*An)*(An)'*Zn;

% find optimum estimate position
   Xe=I*X00+Wn;
   X00=Xe;

% save squar error to stack
  stack((monte+1)/2,1)=((Xe(1,1)-source(1,1)))^2; % x error
  stack((monte+1)/2,2)=((Xe(2,1)-source(1,2)))^2; % y error

  end % end of fetching loop

% find mean square error of 100 monte
   msekf=sum(stack)/100;
```

```
% mean sq.error of 100runs, 40 observations
    ekf(obsv,1)=msekf(1,1);
    ekf(obsv,2)=msekf(1,2);

   if obsv == 1
     EKF2(obsv,1)=ekf(obsv,1);
     EKF2(obsv,2)=ekf(obsv,2);
   else
      buff=sum(ekf)/obsv    ;
      EKF2(obsv,1)=buff(1,1);
      EKF2(obsv,2)=buff(1,2);
 end

end % end of 40 observations

%======================end of Extended Kalman Filter==================
```

2. กรณีที่มี 3 observers

```
% Direction Finding Project
% 3 passive station simulation module
% estimator: EXTENDED KALMAN FILTER

%  find priori estimate position

% Station Location
 source=[15 15];

% find EXTENDED KALMAN FILTER

   for obsv =1:40
    for monte =1:3:300

        X00=[x; y];        % x,y initial
         %Xe=[0;0];         % x,y optimum
          I=[1 0;0 1];      % identity matrix

       Xk1=[0;0];                    % location of sensor1
       Xk2=[station_i(2,1);0];       % location of sensor2
       Xk3=[station_i(3,1);0];       % location of sensor3

  % transfer bearing to new variables

  Zn(1,1)=Zi(obsv,monte)*pi/180;
  Zn(2,1)=Zi(obsv,monte+1)*pi/180;
  Zn(3,1)=Zi(obsv,monte+2)*pi/180;
```

```
% find nonlinear function

        u1=(X00(2,1)-Xk1(2,1))/(X00(1,1)-Xk1(1,1));
        u2=(X00(2,1)-Xk2(2,1))/(X00(1,1)-Xk2(1,1));
        u3=(X00(2,1)-Xk3(2,1))/(X00(1,1)-Xk3(1,1));

        h11=(-u1/(1+u1^2))*(1/(X00(1,1)-Xk1(1,1)));
        h12=(1/(1+u1^2))*(1/(X00(1,1)-Xk1(1,1)));

        h21=(-u2/(1+u2^2))*(1/(X00(1,1)-Xk2(1,1)));
        h22=(1/(1+u2^2))*(1/(X00(1,1)-Xk2(1,1)));

        h31=(-u3/(1+u3^2))*(1/(X00(1,1)-Xk3(1,1)));
        h32=(1/(1+u3^2))*(1/(X00(1,1)-Xk3(1,1)));

     An(1,1)=h11;
     An(1,2)=h12;

     An(2,1)=h21;
     An(2,2)=h22;

     An(3,1)=h31;
     An(3,2)=h32;

     Wn=inv((An)'*An)*(An)'*Zn;

% find optimum estimate position
     Xe=I*X00+Wn;
     X00=Xe;
```

```
% save squar error to stack
    stack((monte+1)/2,1)=((Xe(1,1)-source(1,1)))^2; % x error
    stack((monte+1)/2,2)=((Xe(2,1)-source(1,2)))^2; % y error

    end % end of fetching loop

  % find mean square error of 100 monte

      msekf=sum(stack)/100;

% mean sq.error of 100runs, 40 observations
   ekf(obsv,1)=msekf(1,1);
   ekf(obsv,2)=msekf(1,2);

  if obsv == 1
    EKF3(obsv,1)=ekf(obsv,1);
    EKF3(obsv,2)=ekf(obsv,2);
  else
     buff=sum(ekf)/obsv   ;
     EKF3(obsv,1)=buff(1,1);
     EKF3(obsv,2)=buff(1,2);
   end

end % end of 40 observations

%======================end of Extended Kalman Filter=================
```

3. กรณีที่มี 4 observers

```
% Direction Finding Project
% 4 passive station simulation module
% estimator: EXTENDED KALMAN FILTER

% find priori estimate position

% Station Location
 source=[15 15];


 % find EXTENDED KALMAN FILTER

    for obsv =1:40
     for monte =1:4:400

          X00=[x; y];        % x,y initial
           Xe=[0;0];         % x,y optimum
            I=[1 0;0 1];     % identity matrix

        Xk1=[0;0];                    % location of sensor1
        Xk2=[station_i(2,1);0];       % location of sensor2
        Xk3=[station_i(3,1);0];       % location of sensor3
        Xk4=[station_i(4,1);0];       % location of sensor4

  % transfer bearing to new variables

 Zn(1,1)=Zi(obsv,monte)*pi/180;
 Zn(2,1)=Zi(obsv,monte+1)*pi/180;
 Zn(3,1)=Zi(obsv,monte+2)*pi/180;
 Zn(4,1)=Zi(obsv,monte+3)*pi/180;
```

```
% find nonlinear function

   u1=(X00(2,1)-Xk1(2,1))/(X00(1,1)-Xk1(1,1));
   u2=(X00(2,1)-Xk2(2,1))/(X00(1,1)-Xk2(1,1));
   u3=(X00(2,1)-Xk3(2,1))/(X00(1,1)-Xk3(1,1));
   u4=(X00(2,1)-Xk4(2,1))/(X00(1,1)-Xk4(1,1));

   h11=(-u1/(1+u1^2))*(1/(X00(1,1)-Xk1(1,1)));
   h12=(1/(1+u1^2))*(1/(X00(1,1)-Xk1(1,1)));

   h21=(-u2/(1+u2^2))*(1/(X00(1,1)-Xk2(1,1)));
   h22=(1/(1+u2^2))*(1/(X00(1,1)-Xk2(1,1)));

   h31=(-u3/(1+u3^2))*(1/(X00(1,1)-Xk3(1,1)));
   h32=(1/(1+u3^2))*(1/(X00(1,1)-Xk3(1,1)));

   h41=(-u4/(1+u4^2))*(1/(X00(1,1)-Xk4(1,1)));
   h42=(1/(1+u4^2))*(1/(X00(1,1)-Xk4(1,1)));

An(1,1)=h11;
An(1,2)=h12;

An(2,1)=h21;
An(2,2)=h22;

An(3,1)=h31;
An(3,2)=h32;

An(4,1)=h41;
An(4,2)=h42;

Wn=inv((An)'*An)*(An)'*Zn;
```

```
% find optimum estimate position
      Xe=I*X00+Wn;
      X00=Xe;

% save squar error to stack
    stack((monte+1)/2,1)=((Xe(1,1)-source(1,1)))^2; % x error
    stack((monte+1)/2,2)=((Xe(2,1)-source(1,2)))^2; % y error

    end % end of fetching loop

 % find mean square error of 100 monte

      msekf=sum(stack)/100;

% mean sq.error of 100runs, 40 observations
   ekf(obsv,1)=msekf(1,1);
   ekf(obsv,2)=msekf(1,2);

 if obsv == 1
    EKF4(obsv,1)=ekf(obsv,1);
    EKF4(obsv,2)=ekf(obsv,2);
 else

    buff=sum(ekf)/obsv    ;
    EKF4(obsv,1)=buff(1,1);
    EKF4(obsv,2)=buff(1,2);
  end

end % end of 40 observations

%======================end of Extended Kalman Filter=================
```

ประวัติผู้เขียน

ร้อยเอก สุรเดช เคารพครู เกิดวันที่ 26 มิถุนายน พ.ศ.2511 ที่จังหวัด กรุงเทพมหานคร สำเร็จการศึกษาปริญญาวิทยาศาสตรบัณฑิต (สาขา วิศวกรรมไฟฟ้าสื่อสาร) จากโรงเรียน นายร้อยพระจุลจอมเกล้า รุ่นที่ 38 เมื่อปี พ.ศ.2534 และเข้าศึกษาต่อในหลักสูตร วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า ภาควิชา วิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย เมื่อปี พ.ศ. 2536 ปัจจุบันปฏิบัติราชการเป็น อาจารย์ประจำแผนกวิชาสงครามอิเล็กทรอนิกส์ โรงเรียนทหารสื่อสาร กรมการทหารสื่อสาร เขต ดุสิต กรุงเทพมหานคร