APPENDIX

Appendix

```
clear;
%EN-Modified LEACH%
xm=400;      %diameters of sensor network
ym=400;
sink.x=200;  %distance of base station from the network
sink.y=200;
n = 100;         %no of nodes
p=0.9;          %probability of a node to become cluster head
Eo=0.5;          %energy supplied to each node
ch=n/10;
ETX=50*0.000000001;     %transmiter energy per node
ERX=50*0.000000001;        %reciever energy per mode
Efs=10*0.000000000001;     %amplification energy when d is less than d0
Emp=0.0013*0.000000000001;      %amplification energy  when d is greater than d0
Efs1=Efs/10;   % amp energy just for intra cluster communication.
Emp1=Emp/10;
%Data Aggregation Energy
EDA=5*0.000000001;
a=Eo/2;               %?
rmax=2500;           %no of rounds
%temprature range
tempi=50;
tempf=200;
%Thresholod for transmiting data to the cluster head
h=100;                              %%%%%%Hard Thres%%%%hold H(t)
s=2;                               %%%%%%Soft thres%%%%hold  S(t)
sv=0;                               %%%%%%previously Sensed value S(v)
do=sqrt(Efs/Emp);       %distance between cluster head and base station
do1=sqrt(Efs1/Emp1);
for i=1:1:n
  S(i).xd=rand(1,1)*xm;        %it will distribute the nodes in 1 dimension in x axis
randomly.
  XR(i)=S(i).xd;               %we store its value in xr
  S(i).yd=rand(1,1)*ym;          %it will distribute the nodes in 1 dimension in y axis
randomly
  YR(i)=S(i).yd;
  S(i).G=0;                    % as the no of node that have been cluster head is zero 0
  S(i).E=Eo%%*(1+rand*a);               %?
  %ch.E=x; % initial energy of all cluster heads in network
  %initially there are no cluster heads only nodes
  S(i).type='N';
end

S(n+1).xd=sink.x;   %assume that base station is also a node sp total no of nodes is n
and with base station  it is n+1
S(n+1).yd=sink.y;

countCHs=0;         %the number of Stateflow objects in the current context.
```

Appendix

```
cluster=1;              %first cluster is selected
flag_first_dead=0;
flag_teenth_dead=0;
flag_all_dead=0;
dead=0;
first_dead=0;
teenth_dead=0;
all_dead=0;
allive=n;
%counter for bit transmitted to Bases Station and to Cluster Heads
packets_TO_BS=0;
packets_TO_CH=0;
for r=0:1:rmax
       cv = tempi + (tempf-tempi).*rand(1,1);  %%%%%%Current sensing value C(v)
  if(mod(r, round(1/p) )==0) %remainder
  for i=1:1:n
     S(i).G=0;            % it will assign to the nodes that have not been cluster head .
     %%S(i).cl=0;
  end
  end
dead=0;
for i=1:1:n

  if (S(i).E<=0)
     dead=dead+1;

     if (dead==1)
       if(flag_first_dead==0)
         first_dead=r;
         flag_first_dead=1;
       end
     end

     if(dead==0.1*n)
       if(flag_teenth_dead==0)
         teenth_dead=r;
         flag_teenth_dead=1;
       end
     end
     if(dead==n)
       if(flag_all_dead==0)
         all_dead=r;
         flag_all_dead=1;
       end
     end
  end
  if S(i).E>0
     S(i).type='N';
  end
end
```

Appendix

```
STATISTICS.DEAD(r+1)=dead;
STATISTICS.ALLIVE(r+1)=allive-dead;

countCHs=0;
cluster=1;

if   S(i).type=='C' && S(i).E>a
for j=1:1:ch
   countCHs=countCHs+1;
   S(i).type='C';
        S(i).G=round(1/p)-1;
        C(cluster).xd=S(i).xd;
        C(cluster).yd=S(i).yd;
   distance=sqrt( (S(i).xd-(S(n+1).xd) )^2 + (S(i).yd-(S(n+1).yd) )^2 );
        C(cluster).distance=distance;
        C(cluster).id=i;
        X(cluster)=S(i).xd;
        Y(cluster)=S(i).yd;
        cluster=cluster+1;
distance;

   %   if (cv >= h)
 %test = cv-sv;
 %if (test >= s)
        if (distance>do)
           S(i).E=S(i).E- ( (ETX+EDA)*(4000) +
Emp*4000*(distance*distance*distance*distance ));
        end
        if (distance<=do)
           S(i).E=S(i).E- ( (ETX+EDA)*(4000)  + Efs*4000*(distance * distance ));
        end
        %end

        %packets_TO_BS=packets_TO_BS+1;
        %PACKETS_TO_BS(r+1)=packets_TO_BS;
         %         packets_TO_CH=packets_TO_CH+1;
end
else
for i=1:1:n
  if(S(i).E>0)
  temp_rand=rand;
  if ( (S(i).G)<=0)

      if(temp_rand<= (p/(1-p*mod(r,round(1/p)))))
        countCHs=countCHs+1;
        packets_TO_BS=packets_TO_BS+1;
        PACKETS_TO_BS(r+1)=packets_TO_BS;
         S(i).type='C';
        S(i).G=round(1/p)-1;
        C(cluster).xd=S(i).xd;
```

Appendix

```
          C(cluster).yd=S(i).yd;
         distance=sqrt( (S(i).xd-(S(n+1).xd) )^2 + (S(i).yd-(S(n+1).yd) )^2 );
          C(cluster).distance=distance;
          C(cluster).id=i;
          X(cluster)=S(i).xd;
          Y(cluster)=S(i).yd;
          cluster=cluster+1;

    %   if (cv >= h)
 %test = cv-sv;
 %if (test >= s)

          distance;
          if (distance>do)
             S(i).E=S(i).E- ( (ETX+EDA)*(4000) +
Emp*4000*(distance*distance*distance*distance ));
          end
          if (distance<=do)
             S(i).E=S(i).E- ( (ETX+EDA)*(4000)  + Efs*4000*(distance * distance ));
          end
       end
       end
  % end
   % S(i).G=S(i).G-1;

  end
end
end
STATISTICS.COUNTCHS(r+1)=countCHs;

for i=1:1:n
  if ( S(i).type=='N' && S(i).E>0 )
    if(cluster-1>=1)
      min_dis=Inf;
      min_dis_cluster=0;
      for c=1:1:cluster-1
         temp=min(min_dis,sqrt( (S(i).xd-C(c).xd)^2 + (S(i).yd-C(c).yd)^2 ) );
         if ( temp<min_dis )
            min_dis=temp;
            min_dis_cluster=c;
         end
      end
% if (cv >= h)
 %test = cv-sv;
 %if (test >= s)
          min_dis;
          if (min_dis>do1)
             S(i).E=S(i).E- ( ETX*(4000) + Emp1*4000*( min_dis *min_dis * min_dis
* min_dis));
          end
```

Appendix

```
        if (min_dis<=do1)
            S(i).E=S(i).E- ( ETX*(4000) + Efs1*4000*( min_dis * min_dis));
         end
              S(C(min_dis_cluster).id).E =S(C(min_dis_cluster).id).E- ( (ERX +
EDA)*4000 );
         packets_TO_CH=packets_TO_CH+1;
 %end
  %sv

      S(i).min_dis=min_dis;
      S(i).min_dis_cluster=min_dis_cluster;
   else
      min_dis=sqrt( (S(i).xd-S(n+1).xd)^2 + (S(i).yd-S(n+1).yd)^2 );
         if (min_dis>do)
            S(i).E=S(i).E- ( ETX*(4000) + Emp*4000*( min_dis *min_dis * min_dis *
min_dis));
         end
         if (min_dis<=do)
            S(i).E=S(i).E- ( ETX*(4000) + Efs*4000*( min_dis * min_dis));
         end
         packets_TO_BS=packets_TO_BS+1;

  sv=cv;
    end
 end
end
STATISTICS.PACKETS_TO_CH(r+1)=packets_TO_CH;
STATISTICS.PACKETS_TO_BS(r+1)=packets_TO_BS;
end
first_dead;
teenth_dead;
all_dead;
STATISTICS.DEAD(r+1)
STATISTICS.ALLIVE(r+1)
STATISTICS.PACKETS_TO_CH(r+1)
STATISTICS.PACKETS_TO_BS(r+1)
STATISTICS.COUNTCHS(r+1)

r=0:rmax;
figure (1);
plot(r,STATISTICS.DEAD);
xlabel('Rounds');
ylabel('Dead Nodes');
title('EN-MOD LEACH');
figure (2);
plot(r,STATISTICS.PACKETS_TO_BS);
xlabel('Rounds');
ylabel('Packets to BS');
title('EN-MOD LEACH');
figure (3);
```

Appendix

```
plot(r,STATISTICS.COUNTCHS);
xlabel('Rounds');
ylabel('Number of Cluster Heads');
title('EN-MOD LEACH');
figure (4);
plot(r,STATISTICS.PACKETS_TO_CH);
xlabel('Rounds');
ylabel('Packets to CH')
title('EN-MOD LEACH');
figure (5);
plot(r,STATISTICS.ALLIVE);
xlabel('Rounds');
ylabel('Allive nodes')
title('EN-MODLEACH');
%subplot(2,2,1);
%plot(r,STATISTICS.DEAD);
%xlabel('Rounds');
%ylabel('Dead Nodes ');
%subplot(2,2,2);
%plot(r,STATISTICS.PACKETS_TO_CH);
%xlabel('Rounds');
%ylabel('Packets to CH');
%legend('EN-MOD LEACH ST');   %,'LEACH'
%subplot(2,2,3);
%plot(r,STATISTICS.PACKETS_TO_BS);
%xlabel('Rounds');
%ylabel('Packets to BS');
%subplot(2,2,4);
%plot(r,STATISTICS.COUNTCHS);
%xlabel('Rounds');
%ylabel('Number of Cluster Heads');
%title('\bf LEACH');%
```