American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)

ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

http://asrjetsjournal.org/

Matlab Implementation of Simple Counting Based Weighted Cooperative Spectrum Sensing and Initial Condition Rule

Nuria Ata[a]*, Lee Chaewoo[b]

[a]Employee in INSA and former Graduate of Ajou university, Bisrate Gebriel, Addis Ababa, 124498, Ethiopia

[b]Professor in Ajou University, 206,world cup-roYeongtong-gu,Suwon-si Gyeonggi-do, 443-749, South Korea

[a]Email: qemer2030@gmail.com

[b]Email: cwlee@ajou.ac.kr

Abstract

Cooperative spectrum sensing allows strict regulatory performance requirement to be relaxed on local sensing. In practice secondary users are more likely to experience distinct signal strength depending on distance from primary transmitter. This shows the need for weighting local decision by local reliability. In this paper we discuss implementation issue of simple counting based decision weighting method. And we provide solution and a complete MATLAB implementation code. We demonstrate, by carefully selecting the initial conditions, we can get stable performance. And also our results shows that the optimal weighted method outperforms the existing equal weight combining in terms of lower total error probability.

Keywords: Secondary user; optimal weight; simple counting rule; error probability; initial condition rule.

1. Introduction

Traditional spectrum sharing method, Figure 1, allocated entire radio spectrum band to different licensed organizations for permanent use and it reserved small portion- the ISM band for low power devices and other electronic devices. However spectrum congestion has become increasingly problematic in the ISM band because of the emerging new technologies, the growing demand for big data, and smart city. Which are all dependent on wireless communication.

----

* Corresponding author.

Surprisingly studies conducted on spectrum usage in different countries, have revealed that most of the licensed spectrum is not used efficiently because the licensed transmitter does not use the spectrum all the time. Cognitive Radio (CR) technology provides efficient spectrum sharing solution. At the heart of cognitive radio technology is spectrum sensing algorithm which enables Secondary Users (SU) to identify vacant licensed spectrum.

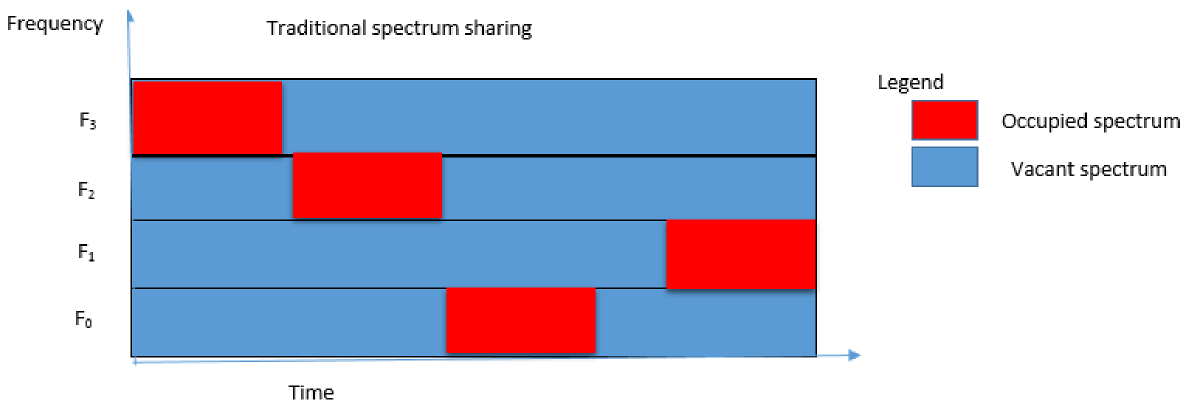

Figure 1: spectrum activity of primary user (PU)

There are various sensing algorithms, energy detection being the simplest method and the most widely used. However, its performance is dependent on received signal strength. As shown in Figure 2, the SU makes its decision based on the sensing result. When the sensing result is '1', it avoids transmission to allow the PU continue its transmission undisturbed. And when the sensing result is a '0', the SU transmits to utilize the unoccupied spectrum. However there are a number of factors that could reverse the sensing result, for instance in the figure below missed detection occurred in the first sensing period, this wrong decision caused the SU to transmit while the licensed channel is being used by the PU. Another case is when the SU fails to utilize vacant spectrum due to a false alarm error. The problem of spectrum sensing is therefore a tradeoff between minimizing interference and maximizing spectrum efficiency.

Minimizing interference on PU, largely depends on the received strength, whereas spectrum efficiency is a function of the threshold position as shown in Figure 3 and 4. When the received signal is stronger, the two distributions i.e. H0-PU idle and H1-PU active is large Figure 3, in this case we can increase the threshold to minimize the occurrence of false alarm. However, if the received signal is weak as in Figure 4, the two distributions overlap and we are forced to lower the threshold to minimize the missed detection probability and this action increases the false alarm probability. For this reason threshold selection is a tradeoff between detection probability and false alarm. Although energy detection is a simple method, its performance is lower compared to other methods.

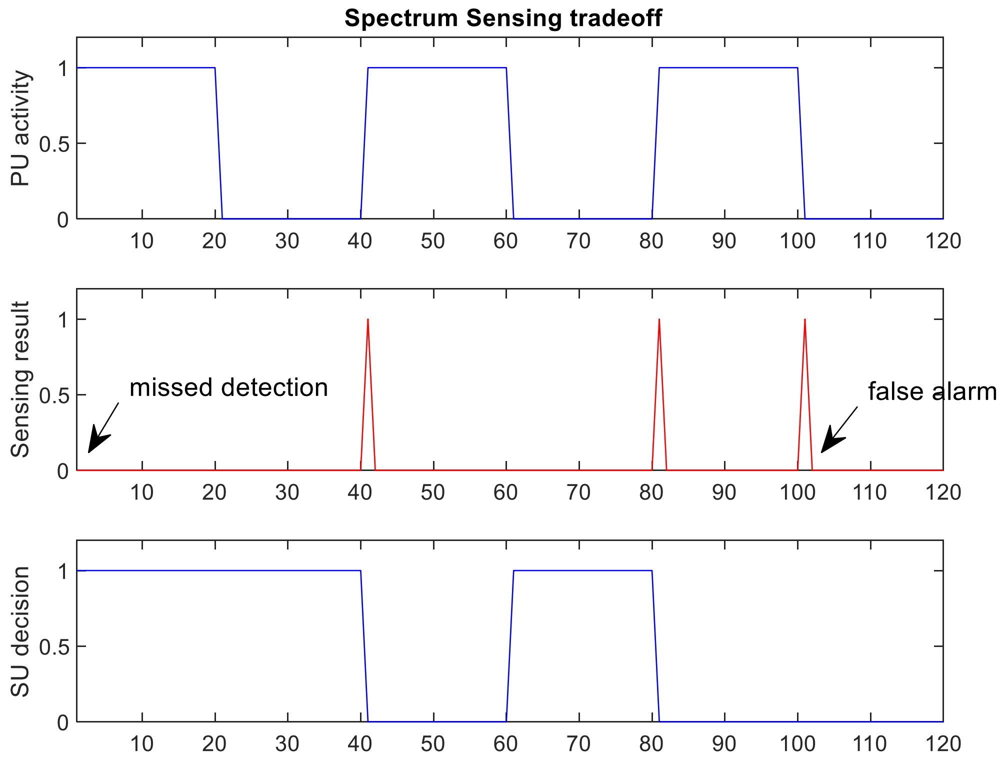

Figure 2: spectrum sensing for detecting and using vacant spectrum

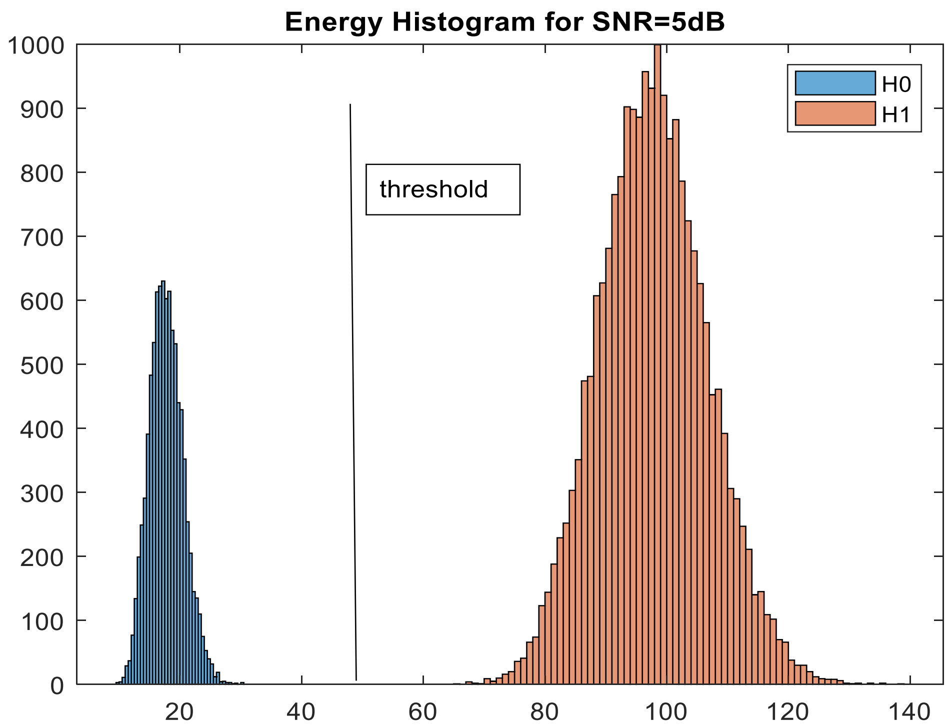

Figure 3: Histogram of H0 and H1 for high SNR

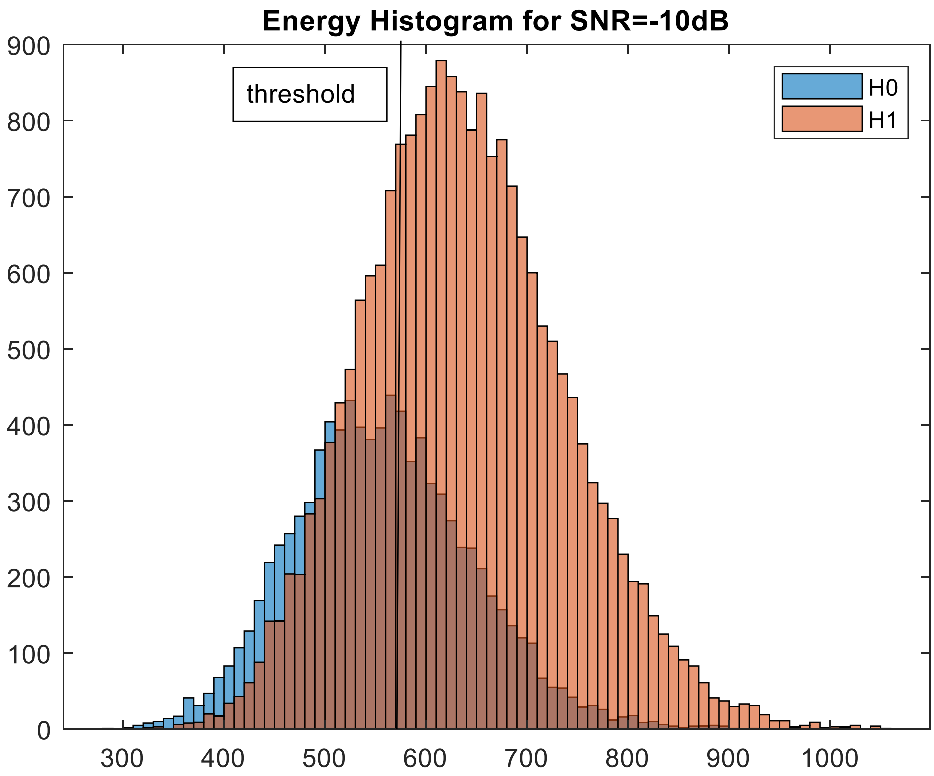

Figure 4: Histogram of H0 and H1 for low SNR

The strict regulatory performance requirement cannot be meet by a single SU. Cooperative Spectrum Sensing (CSS), came to overcome this problem. CSS can be performed within a single hop or multi hop. In CSS, the local sensing decision of different CR is gathered and combined to make a global decision. This final decision is broadcasted to all the SU. All CR or SU make transmission decision based on it. In practice depending on CR relative position to the PU, it is expected that each CR observes different signal strength. Especially the most distant SU receives very weak signal. Therefore it is necessary to take this local reliability in to account when doing CSS. The problem of weighting local decisions is how to determine the local sensing reliability without prior information or with some information. There have been proposed various versions of combining methods such as the equal weight combining, Maximal ratio combining, and many others. Authors in [10] proposed optimal weighting algorithm in which the Fusion Center (FC) gradually adapts the local sensing reliability of each CR based on observation. In this paper we analyze this algorithm and assess the implantation issues. We set conditions for determining the initial value of the parameters for stable performance. We compare the total error probability of this method with the equal weight combining method. Our results show, if the initial conditions are set properly, the algorithm gives stable performance and converges faster. We also give MATLAB code for the algorithm. The rest of the paper is organized as follows: Section 2 provides the research methodology. Section 3 presents simulation setting. The result and discussion is presented in Section 4. Followed by conclusion in Section 5 and finally the MATLAB code is presented in the end.

2. Methodology

- There are different ways of measuring local sensing reliability. Authors in [9] proved the optimal weight of the i^{th} SU is a function of its local performance. However there is no prior information on the local detection probability p_{di} , false alarm probability p_{fi} , correct rejection probability p_{ri} and the missed-detection probability p_{mi} in the fusion center. Authors in [10] proposed a simple counting rule for estimating these local performances. In this paper we analyzed implementation issue of the simple counting rule and in this process we discovered a rule for properly initializing the parameters. And as a result we are able to resolve the problem in initializing the algorithm and we are able to achieve a more stable and reliable performance.
- Let weights b_i and c_i denote the i^{th} SU local sensing reliability to minimize false alarm and misdetection error probabilities respectively. According to authors in [10], the optimal data fusion is given as follows:

$$D_g(k) = \begin{cases} 1, \; if \left(a_0 + \sum_{i=1}^{L} b_i D_i(k) - \sum_{i=1}^{L} c_i \bar{D}_i(k) \right) > 0 \\ 0, otherwise \end{cases} \quad (1)$$

Where the weight parameters are estimated using formulas (2) and (3).

$$\begin{cases} b_i = \log \dfrac{P_{d_i}}{P_{f_i}} \\ c_i = \log \dfrac{1 - P_{f_i}}{1 - P_{d_i}} \end{cases} \quad (2)$$

$$a_0 = \log \frac{\Pr(H_1)}{\Pr(H_0)} \quad (3)$$

- In equation (1), D_i stands for local decision of the i^{th} SU, D_g is the global decision. H_1, H_0 represent PU ON and PU OFF state respectively.
- To build our implementation idea we made the following facts as a foundation for our analysis of the algorithm:

a) The weight b_i which is a function of correct detection and false alarm probabilities must always be positive $b_i = \log \frac{pd_i}{pf_i} > 0$ because the detection probability is always larger than the false alarm probability.

b) The weight c_i must always be positive $c_i = \log \frac{pr_i}{pm_i} > 0$ because the correct rejection probability is larger than missed-detection probability. To implement the simple counting rule.

- let x be count of PU idle

- let y be count of PU ON
- let m_i, f_i, d_i, r_i be misdetection, false alarm, correct detection and correct rejection counts of the i^{th} SU respectively. Therefore the formula in (2) and (3) can be approximated as in equation (4) and (5).

$$\begin{cases} b_i = \log\frac{d_i}{f_i} - \log\frac{y}{x} \\ c_i = \log\frac{r_i}{m_i} + \log\frac{y}{x} \end{cases} \qquad (4)$$

$$a_0 = \log\frac{y}{x} \qquad (5)$$

- Initially we set all these variables to '1'. However, this led to unstable performance in the simple counting rule. And it was frequently showing inversion in the global decision which makes the total error probability close to one. After carefully examining the unexpected result, we were able to observe that the weights were becoming highly negative and were approaching to negative infinity. And this meant that decisions where misclassified. For example when PU was active, global decision indicated a false alarm and when PU was idle then the global decision was classified as missed detection. And similarly a false alarm was globally marked as correct detection and the missed-detection was classified as a correct rejection.
- Our analysis showed there is no any condition that the weight becomes negative. Because if our implementation is correct then the facts in (a) and (b) must be met. So we came in to conclusion that our initialization was wrong. Drawing from this analysis we made the following rule to initialize the parameters correctly.

  a) Assumption: the probability PU occupying its spectrum is higher than idle state. Therefore $y > x$

  b) The counts d_i, r_i can be set higher initially , $d_i > r_i$ because PU active state is highly probable over idle state

  c) The counts f_i, m_i can be set lower because these error probabilities are lower than d_i and r_i. Also $f_i > m_i$

  d) Initially which SU has high SNR is unknown, so the initial condition need to be identical for all SUs and gradually as the algorithm observes the SUs, it adjusts their performance

3. Simulation

We limited number of secondary users to 4. First we determined optimal threshold for different SNR values. We generated PU active state message with probability of 0.7 and therefore the PU inactive is 0.3. We assumed AWGN channel condition for the path between PU and SU. The sensing period is set to 10ms for 8 KHz sampling rate. In the receivers (SU), white noise was added to the received signal. Each SU locally measures received signal energy and compares it against a fixed optimal threshold. These local decisions are gathered at the FC and combined by weighing with the estimated reliability. All parameters are set to some values according to our initialization rule, and the initial reliability is estimated from these initial values. Then by observing current performance the FC updates reliability for each SU. We estimated weight for each SU and we computed the weight estimation error. The weight estimation error with respect to number of sensing period is plotted. We also plot, estimated weight with number of sensing period. We evaluated the weighted CSS and compared it with equal gain CSS.

4. Results and Discussion

Result in Table 1 first row, illustrates when SUs has distinct SNR, higher sensing performance can be achieved by using the weighted cooperative sensing (WCSS). Which means if experiment is conducted for 100,000 times, the WCSS will make total 13260 error, which is 7300 times lower than the majority rule based equal gain combining (EGC). This performance improvement leads to a more efficient spectrum utilization, while minimizing interference on PU. When SNR is kept identical for all users, the outcome is similar. In practice users experience different signal strength therefore by using optimal weighting the total error probability can be reduced significantly. The data in table 2 shows the ability of the simple counting rule to adapt SU local performance. The method assigns highest weight to 20dB SNR and lowest value is given to -15dB SNR. The interpretation of this is that the global sensing decision is more impacted by the stronger local signal and this minimizes the chances of erroneous final decision, which occurs due to less reliable users. Figures 5 and 6, demonstrate the weight estimation error decaying when the weight converges to the ideal value. The global decision is more reliable over local decision, for this reason it is used to judge the performance of each user. Initially all SU have identical performance, as the fusion center is able to get more information on the behavior of each user, it adjusts the reliability. The decaying nature of the weight estimation error proves the ability of the fusion center to assign reliability closer to the actual local performance. In this case, because all SU have identical SNR, the weight estimation error for all SU is the same. In Figures 7 and 8 we can see that the three SUs has low reliability and thus the weight estimation error decays close to ‘0’. This shows that the impact of low reliability SU is reduced faster, allowing the fusion center to make more reliable decision. The gap between the weight estimation error of the 20dB SU and the other three SUs proves the majority of the global decision is influenced by this user. Although in the beginning all users have equal influence, gradually the fusion center is able to learn and give more credibility to the data from the highly reliable user, when making the final decision. Note, ideal weight refers to the local reliability which is obtained from knowledge of local performance. On the other hand the estimated weight refers to the reliability value obtained by approximating local performance probabilities using counting method. Finally Figures 9 and 10 demonstrate how the estimated weight gradually increases to reach the desired local sensing performance over the number of sensing experiment. These result show the weight learning process converges faster for low SNR.

Table 1: Weighted cooperative sensing vs Equal gain

SNR and threshold	Total error probability for WCSS	Total error probability for equal gain CSS
SNR in dB=[20, -15, -10,-12] Thres=[40, 1700, 595, 930]	0.1326	0.2056
SNR set to -5dB for all SUs Threshold set to 210	0.0334	0.0408

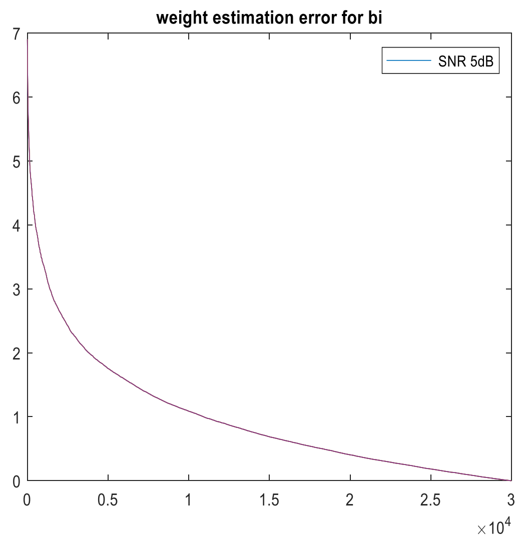

Figure 5: weight estimation error for bi, identical SNR

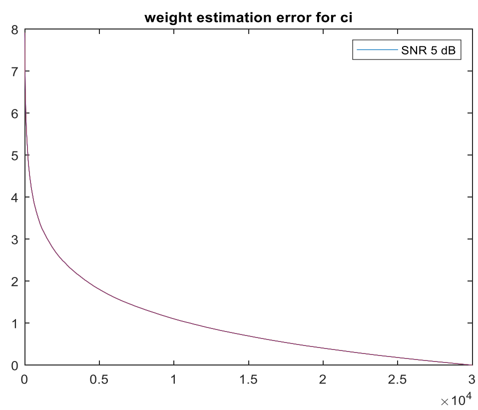

Figure 6: weight estimation error ci, identical SNR

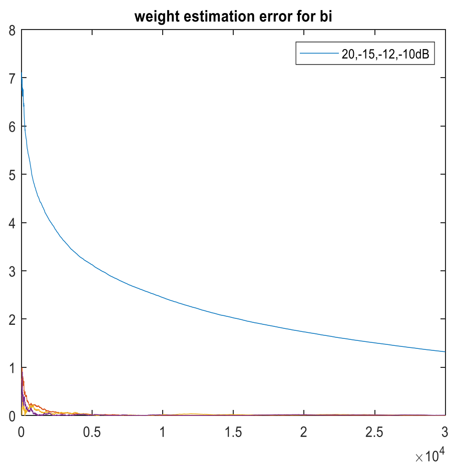

Figure 7: bi estimation error for different SNR

Table 2: Estimated weight for identical and different SNR

SNR in dB and Threshold	Estimated weight
SNR in dB=[20, -15, -12, -10] Thres=[40, 1700, 930, 595]	bi=[3.4465, 0.1703, 0.5160, 0.7281] ci=[5.6365, 0.2986, 0.4687, 0.7162]
SNR =[5, 5,5,5] Thres=40	bi=[7.7274, 7.7274, 7.7274,7.7274] ci=[9.2560, 9.2560, 9.2560, 9.2560]

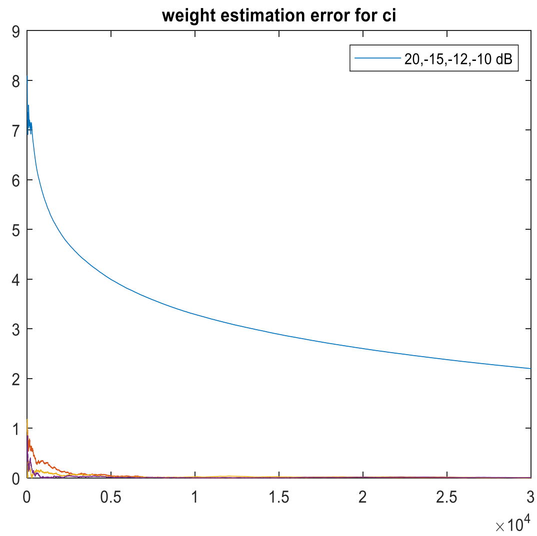

Figure 8: ci weight estimation error for different SNR

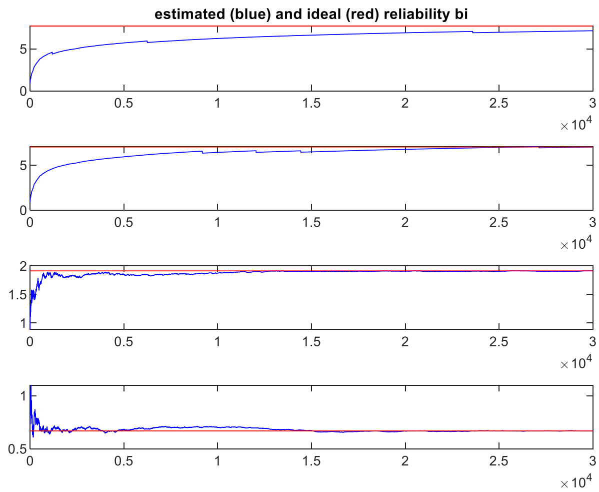

Figure 9: estimated weight bi for snr=[5 0 -5 -10] dB

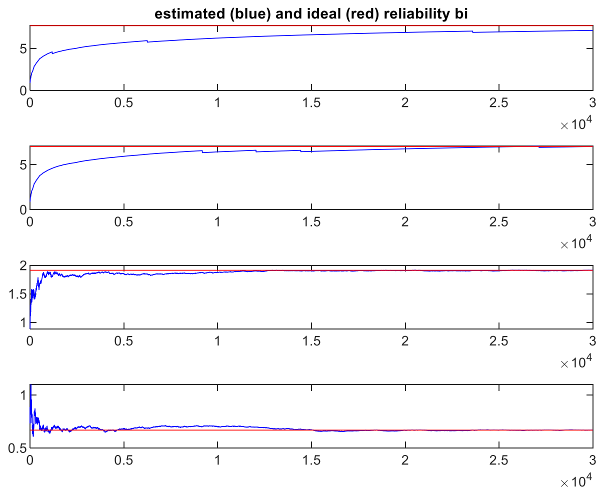

Figure 10: estimated weight ci for SNR= [5 0 -5 -10] dB

5. Conclusion

By carefully initializing the parameters, the weight estimation algorithm is able to give reliable and stable result. And also converges faster. We have been able to demonstrate how total error probability of cooperative sensing can be reduced by employing local sensing reliability. The adaptive weight estimation algorithm is able to predict local sensing reliability based on observation of local sensing decisions of each SU. In our next paper, we would like to work on showing the conditions for effective optimal weighting method. And also we will work to improve stability and convergence of this method.

Acknowledgments

This research is funded by Basic Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2014R1A1A2057567).

6. MATLAB Code

```
close all;clc;clear;

nSU=4;fs=8000;ts=0.01;%num of secondary users,%sampling rate,%sensing period

M=ts*fs;Nf=30000;%number of samples per frame,%number of primary frames

xp=zeros(1,Nf*M);P_ON=0;% propability the PU is active

P_state=zeros(1,Nf); %PU transmission state

%Step#1: Generate primary user signal
```

```
for i=1:Nf    %generate primary user data

    if(rand(1)<=0.7)

       xp((i-1)*M+1:i*M)=sign(randn(1,M));

       P_ON=P_ON+1;P_state(i)=1;

    end

end

P_OFF=Nf-P_ON;snrdB=[5 0 -5 -10];y_su=zeros(nSU,Nf*M);

%Step#2: Model received signal under awgn channel for secondary users(SU)

for m=1:nSU

    y_su(m,:)=awgn(xp,snrdB(m),'measured');% awgn model of received signal at mth SU

end

%Step#3:  Local Sensing, Determine threshold

E_su=zeros(P_ON,nSU);  %SUs measured energy for active PU

D_su=zeros(nSU,Nf);thres=[40 90 210 595];

for i=1:Nf

    for m=1:nSU

     E_su(i,m)=sum( y_su(m,(i-1)*M+1:i*M).^2);

     D_su(m,i)=E_su(i,m)>=thres(m);

    end

end

id=15;icr=7; ifa=4; imd=2;ny=id+imd; nx=icr+ifa;

ncd_su=id*ones(2,nSU);%ideal and measured correct detection counts
```

```
ncr_su=icr*ones(2,nSU);%ideal and measured correct rejection counts

nfa_su=ifa*ones(2,nSU);%ideal and measured false alarm counts

nmd_su=imd*ones(2,nSU);%ideal and measured missed detection counts

 %Step#4: Weighted Cooperative Spectrum sensing

b_est=(log(id/ifa)-log(ny/nx))*ones(1,nSU);c_est=(log(icr/imd)+log(ny/nx))*ones(1,nSU);   %estimated   weight
SUs

npu_on=ny; npu_off=nx; %estimated number of pu on and off

De=zeros(1,Nf);  %equal weight cooperative sensing decision

b_arr=zeros(nSU,Nf);c_arr=zeros(nSU,Nf); %wieght arr for visualization

th=[0,2];a0=0;

Dw_id=[zeros(1,Nf);P_state];%wieghted decision and ideal states of PU

for i=1:Nf

   Dm=sum( D_su(:,i));

   Dg= a0+sum(b_est.*D_su(:,i)')-sum(c_est.*(1-D_su(:,i))');

   d=[Dg,Dm]>=th; Dw_id(1,i)=d(1);De(i)=d(2);npu_on=npu_on+d(1);

   for m=1:nSU

      b_arr(m,i)=b_est(m);c_arr(m,i)=c_est(m);

   end

   if(Dw_id(1,i)==0)

      npu_off=npu_off+1;

   end

   for k=1:2  %counting

      for m=1:nSU
```

```
            if((D_su(m,i)==1) && (Dw_id(k,i)==1))
                ncd_su(k,m)= ncd_su(k,m)+1;
            elseif((D_su(m,i)==0) && (Dw_id(k,i)==1))
                nmd_su(k,m)=nmd_su(k,m)+1;
            elseif((D_su(m,i)==0) && (Dw_id(k,i)==0))
                ncr_su(k,m)=ncr_su(k,m)+1;
            else
                nfa_su(k,m)=nfa_su(k,m)+1;
            end
            b_est(m)= log(ncd_su(1,m)/nfa_su(1,m))-log(npu_on/npu_off);
            c_est(m)= log(ncr_su(1,m)/nmd_su(1,m))+log(npu_on/npu_off);
        end
    end
    a0=log(npu_on/npu_off);
end
ncd_su= ncd_su-id;ncr_su=ncr_su-icr;
nfa_su(1,:)=nfa_su(1,:)-ifa;nmd_su(1,:)=nmd_su(1,:)-imd;
npu_off=npu_off-nx;npu_on=npu_on-ny;
pd=zeros(1,2);pmd=zeros(1,2);pfa=zeros(1,2);pr=zeros(1,2);D=[Dw_id(1,:);De];
for i=1:Nf
    for m=1:2
        if(P_state(i)==1&& D(m,i)==1)
```

```
         pd(m)=pd(m)+1;
      elseif(P_state(i)==1&&D(m,i)==0)
         pmd(m)=pmd(m)+1;
      elseif(P_state(i)==0&&D(m,i)==1)
         pfa(m)=pfa(m)+1;
      else
         pr(m)=pr(m)+1;
      end
   end
end
pd=pd/P_ON;pmd=pmd/P_ON;pr=pr/P_OFF;pfa=pfa/P_OFF;
p_tot_w=1/2*(pfa(1)+pmd(1));p_tor_e=1/2*(pfa(2)+pmd(2));
%Step#5: Demonstrate Result
 t=0:Nf-1;figure(1)
for m=1:nSU
   b_id=ones(1,Nf)*(log(ncd_su(2,m)/nfa_su(2,m))-log(P_ON/P_OFF));
%        plot(t,b_arr(m,:), 'b',t,b_id,'r');legend('b estimate,ideal');hold on;
    plot(t,abs(b_id-b_arr(m,:)));legend('error for bi');hold on;
end
 figure(2)
for m=1:nSU
 c_id=ones(1,Nf)*(log(ncr_su(2,m)/nmd_su(2,m))+log(P_ON/P_OFF));
```

```
% subplot(nSU,1,m);

%     plot(t,c_arr(m,:), 'b',t,c_id,'r');legend('c estimate, ideal');hold on;

plot(t,abs(c_id-c_arr(m,:)));legend('error for ci');hold on;

end
```

References

[1]. P. K. Verma , S. Taluja , R. L. Dua.“Performance analysis of energy detection, matched filter detection & cyclostationray feature detection spectrum sensing techniques.” IJCER, vol.2, Sep. 2012.

[2]. A. Ghasemi, E. S. Sousa.” Spectrum sensing in cognitive radio networks: requirements, challenges and design trade-offs. “ Commun. Magazine, IEEE, pp.32-39, vol.46, Apr. 2008.

[3]. D. Sun , T. Song , M. Wu, J. Hu , J. Guo , B. Gu.“Optimal sensing time of soft decision cooperative spectrum sensing in cognitive radio networks”, 2013 IEEE WCNC, pp. 4124-4128, Apr. 2013.

[4]. N. Nguyen-Thanh, I. Koo.” Acluster-based selective cooperative spectrum sensing scheme in cognitive radio.” EURASIP J. Wireless. Commun. Networking, Jun. 2013.

[5]. X. Huang, N. Han, G. Zheng, S. Sohn. “Weighted-collaborative spectrum sensing in cognitive radio.“ Commun. On Networking, China, pp.110-114, Aug. 2007.

[6]. S. A. Alvi, M. S. Younis, M. Imran, Fazal-e-Amin. “A weighted linear combining scheme for cooperative spectrum sensing.” International Conference on Ambient Systems, Networks and Technology, pp.149-157, vol. 32, Jun. 2014.

[7]. L. Xiao, K. Liu, and L. Ma.“A weighted cooperative spectrum sensing in cognitive radio networks” , ICINA, pp. v2-45 - v2-48, vol. 2, Oct. 2010.

[8]. S. Maleki, S. P. Chepuri, G. Leus.“Energy and throughput efficient strategies for cooperative spectrum sensing in cognitive radios.” SPAWC, 2011 IEEE 12th International Workshop on, pp.71-75.

[9]. Z. Chair, P. K. Varshney.“Optimal data fusion in multiple sensor detection systems.” IEEE Transactions on, Aerospace and Electronic Systems, vol. AES-22, pp.98-101, Jan.1986.

[10]. N. Mansouri, M. Fathi.“Simple counting rule for optimal data fusion.” Proc. IEEE. Conf. on Control Applications, vol.2, pp.1186-1191, Jun 2003.