Chapter 6

Interference Detection Techniques

6.1 Introduction

Radio Frequency Interference(RFI) is a common problem during spectral line observations. This RFI can be edited manually by inspecting the individual power spectra. RFI lines can appear at various positions in the spectrum depending upon the nature of the interference or the instrument settings. Here a method of detection has been described which applies to the power spectrum(i,e after the fourier transform of the recorded data) so that the RFI can be detected and edited using an algorithm and the astronomer is completely releaved of monitoring the power spectrum. Using such a RFI detection algorithm one can analyse huge amounts of data both with minimum time and effort. The method has been described as it applies to radio spectral line observations using dual Dicke switching. This is the technique which was used to observe spectral lines with ORT as well as WSRT.

6.2 Radio Spectral Line Observations

In a typical radio spectral line observation one first decides upon the frequency at which the line will be observed. The bandwidth required to detect the line or lines and the period for which the observation will be made. This period decides the signal to noise ratio in the final spectrum. The line is assumed to be observed with the dual Dicke switching. In this observation the band is split into two parts, once the spectral line is made to appear in one part and next in the other part. This is usually done by appropriately selecting two fixed local oscillator frequencies over a period of time so that the gain characteristics of the instrument do not change much. This is done alternately all along the observation. Hence the observation consists of two types of spectra one with the spectral line appearing in the right part & the other with the spectral line in the left part. The first is called the T_{on} spectrum & the other will be called the T_{off} spectrum(this is just a convention). The power spectrum P basically is

written as,

$$P = Gain \times (noise + signal). \qquad (6.1)$$

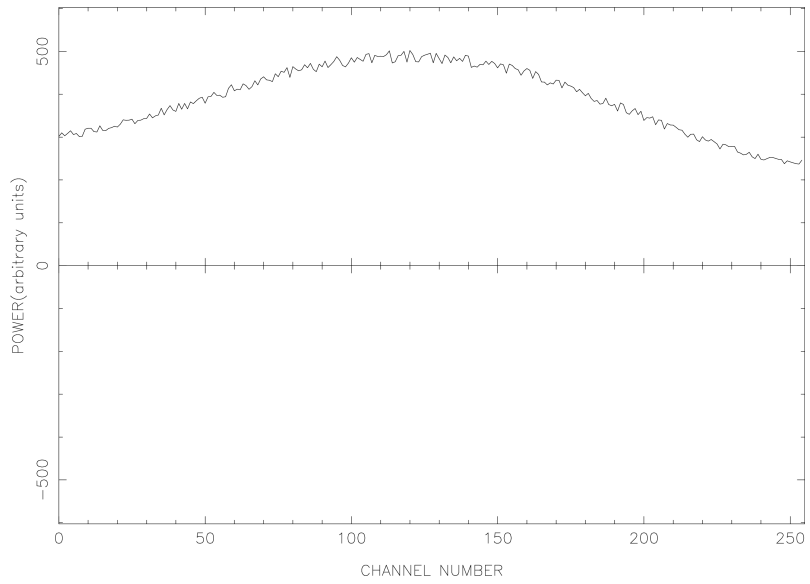

Fig 6-1: A typical power spectrum in radio spectral line observation.

Noise has both the system noise as well as the noise incident on the telescope along with the signal. A typical power spectrum has the appearance as shown in the fig-1. PS consists of two lines, one from T_{on} and the other from T_{off}. The spectral line from T_{off} will be inverted since -ve of T_{off} was added to T_{on}. The rest of the band is assumed to be gaussian noise(or approximately gaussian) with an inherent baseline. Since there is no difference between addition or substraction of gaussian noise due to its symmetry. Both lead to gaussian noise again. After this PS is folded and averaged appropriately so that the inverted line overlaps with the one in the other part. Ofcourse a negative of the inverted part will be added. This completes the simple data processing. Many such spectra can be added to improve the signal to noise ratio which eventually results in spectral line detection. It is the property of gaussian noise that the rms(root mean square) of the averaged spectrum reduces by a factor of $1/\sqrt{n}$ when n spectra are averaged.

6.3 Interference

RFI is a common enemy of spectral line observation. It can be produced in a

variety of ways, by electrical sparking, computers, electronic gadgets etc. The magnitude of interference can be both small as well as large. It is easy to detect the large interference while the small interference is the one which is to be tackled. A typical interference infected spectrum is as shown in the fig-2(some times it is low to a level that it is distinguishable only in the T_{on}/T_{off} - 1 .

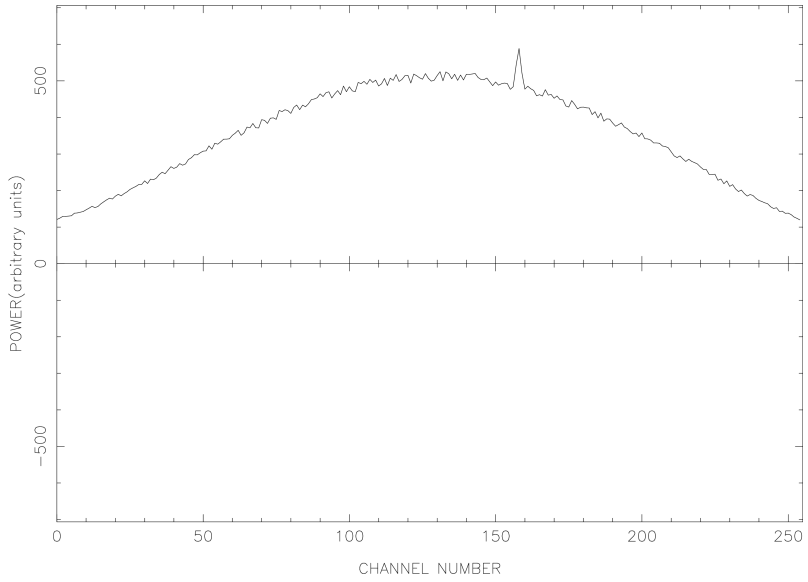

Fig 6-2: A typical interference infected power spectrum. for the same of simplicity its magnitude has been kept small. In reality the interference could be both very small, indistinguishable on the powerspectrum as above or very large than this.

Different sources produce different types of RFI. For example the RFI generated by electrical sparking will produce a series of lines. In such cases the standard deviation of the data itself may increase drastically. Using this information such data can be discarded. While that produced by a computer or an electronic gadget may be a single or multiple number of lines at some fixed position or positions in the band. In such cases one has to devise RFI excision techniques.

6.4 RFI Detection

The RFI lines appearing in the power spectrum can be eliminated manually by carefully inspecting it. But this is labourious and time consuming. Here a method to

detect interference will be explained which can be incorporated as an algorithm in the analysis program. By an appropriate code one can also have them edited(this is left to the user) out as per the choice and requirements. This reduces manual labor as well as results in speedy data anaysis. In the method proposed here the following assumptions have been made

- The RFI can be observed in a short integration time where as the detection of the astronomical signal would require a much longer integration time. It means that if the spectrum be subject to interference detection, where as the interference is above the limits set(explained later). But the astronomical line is weak enough(within the expectation limits) or comparable with the standard deviation of the noise in the spectrum.
- The RFI is narrow band, i.e only a small portion or portions of the band are infected with it. It is necessary to use a portion of the band to find the standard deviation of the spectrum. The parameter is necessary in the detection of interference. The standard deviation is an important quantity relevant to noise.
- It is also not necessary that the RFI be time variable. If the spectrum has interference as per the technique used here the associated channels will be flagged or else they will pass unflagged.

The technique to detect RFI depends on the properties of gaussian noise. When pure gaussian noise across N channels is considered there exists an expected maximum value that can occur across the channels. Similarly there exists for N channels a maximum for the difference between two adjacent channels. Lastly another quantity of interest is the added difference which is the sum of consecutive differences of similar sign. Suppose these difference signs go like + - + - + + - , then the fifth and sixth differences will be added to form one quantity. These quantities can be calculated theoretically and used to inspect the gaussian noise. Typical interference being sharp and narrow would produce, large values or differences between adjacent channels, that deviates considerably from the expected values. On such an occurence the corresponding channel or channels can be flagged for interference. The first quantity, expected maximum can be of use only when the spectrum has reasonably zero mean all across the spectrum, and so is of less importance. The other two quantities are free from this defect as they are differences between adjacent channels which more or less have the same DC value. The methods using these three quantities have been named as

direct,difference and added difference methods. The added difference traps those interference lines that escape the difference method due to their being broad. Plots of these quantities signifying there use to detect interference have been given in the following sections. Practical results of applications on ORT and WSRT data have also been displayed at the end of the chapter.

6.4.1 Direct Method

This is the simplest of all the methods. In this method the spectrum of N_{data} channels is divided into n_{blk} number of blocks. Each block has now

$$N_{blk} = \frac{N_{data}}{n_{blk}} \tag{6.2}$$

number of data points. The block with the minimum standard deviation will be considered to be the representative of good data, meaning to say free of any interference. This is consistent in the sense that in the regions without any signal one has only noise, any interference introduced into this data will increase the standard deviation. The minimum standard deviation block will be used as a reference to calculate the required parameters to detect interference as well as to edit and repair it(one can replace the edited portion with noise of same standard deviation as that in the reference block). The absolute maximum value in this block will be used to generate a limit on the maximum absolute value that could be encountered in the other blocks by choosing a suitable multiplicative factor. This can also be done by calculating the standard deviation in the reference block. The expected absolute maximum is directly proportional to the standard deviation. Use this standard deviation and the total number of channels to calculate the expected maximum using the expression given for the expected maximum(X_{mx}) eqn (6.4). It should be noted that a constant muliplicative factor(>1, say 1.2 or 1.3) has to be used with the expected maximum value to allow for the realistic departures from the expected value. If any block has any channel or channels with absolute value greater than this, then those channels in that block can be flagged for interference. The expected maximum absolute value for a N channel gaussian noise with zero mean is given by

$$< X_{mx} > \; = \; \sigma \sqrt{2} \int_0^1 erf^{-1} \, s^{1/N} \; ds \tag{6.3}$$

which has a convenient approximation as,

$$< X_{mx} > \; \approx \; \sigma \sqrt{2} \; \; erf^{-1} \left[1 \; - \; \frac{0.5833}{N} \right] \tag{6.4}$$

Derivation of this has been discussed in the appendix-II. This method requires that the spectrum under investigation be symmetrically distributed about the zero line. This can be brought about by calculating the mean of all the channels of the data and substracting it from each channel value of the spectrum. Also in some cases a baseline should be removed from the spectrum before subjecting to RFI detection. In the case where the interference is present this is difficult since the fitted curve will be biased by the interference line. If one is sure that the spectrum has no baseline then one can use this method or else omit it. However the difference method is more tolerant towards this.

6.4.2 Difference Method

The difference method is a simple method of interference detection. In this method adjacent channel differences all across a single spectrum will be found. For a N channel gaussian noise there exists an upper limit on the absolute maximum expected difference. Which will be used as a reference to constrain the maximum value that could be encountered across the N channels. Further the spectrum is divided into n_{blk} blocks. Again the reference block will be the minimum standard deviation block. In each block the adjacent differences are found. A typical plot of differences of an interference infected T_{on}/T_{off} - 1 spectrum is shown in the fig-3. Obviously the differences in the interference infected channels are much higher than the differences in the other channels. The expected value of the maximum difference for gaussian noise with N data points is given by the relation,

$$\begin{aligned} < D_{mx} > \; &\approx \; 2\sigma \int_0^1 erf^{-1} \, s^{1/(N-1)} \; ds \\ &\approx \; 2\sigma \; \; erf^{-1} \left[1 \; - \; \frac{0.5833}{N} \right] \end{aligned} \tag{6.5}$$

where σ is from the reference block. The derivation of this has been discussed in the appendix-II. In practice a multiplicative factor > 1 (say 1.2 or 1.3) has to be chosen to allow for realistic deviations from the expected value. As it never happens that if the expected value is D_{mx} one will always see a value less than D_{mx}. Both higher and lower values are possible. The multiplicative term(>1) allows a margin for the higher random values.

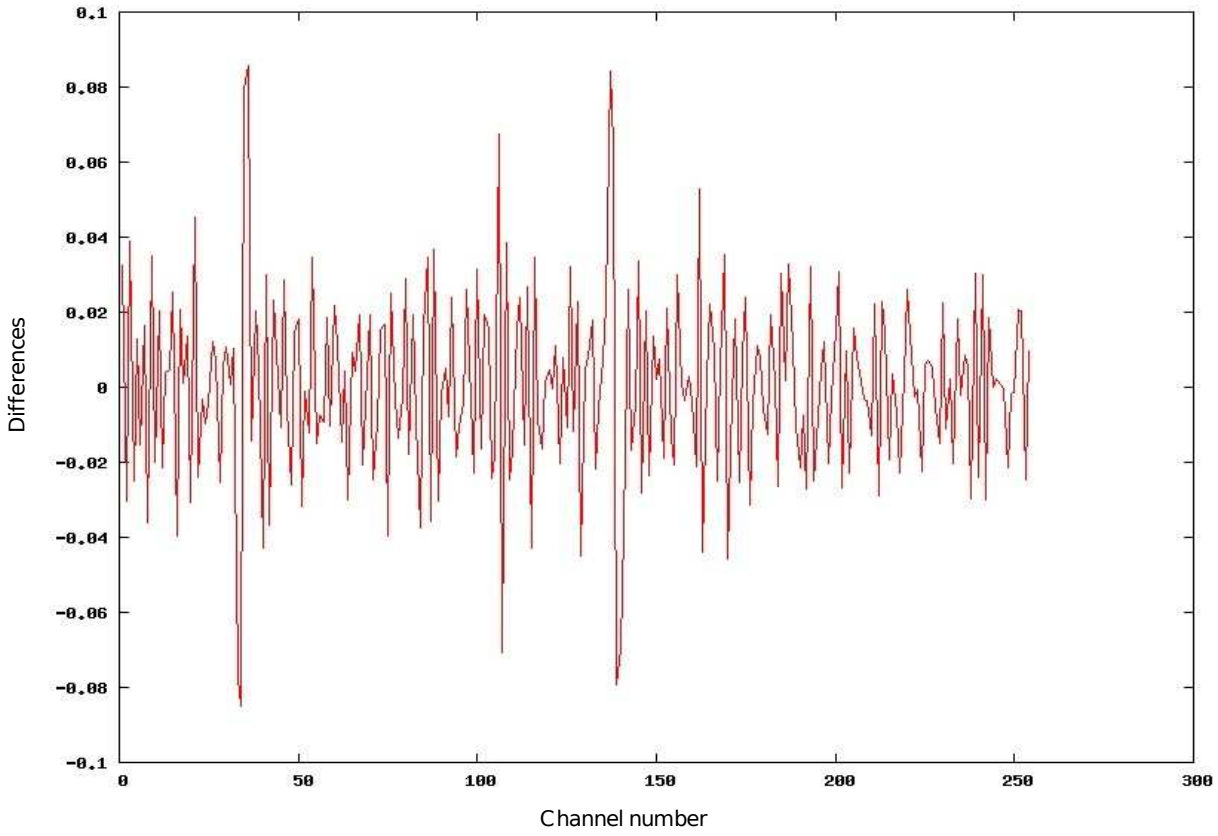

Fig 6-3: The adjacent channel differences for a interference infected spectrum. Notice the large differences at the channels infected with interference.

6.4.3 Added Difference Method

The added difference method is similar to the difference method except now the consecutive differences with same sign will be added to form a new array of quantities. Suppose the difference signs go like + - + - + + - , then the fifth and sixth differences will be added to form one quantity. The plot of such an array for a typical interference infected spectrum is shown in the fig-4. The improvement of interference detection criteria is clearly visible in this plot. Added difference method traps those interference

lines which escape the difference method due to their being broad. This method is more sensitive than the difference method but it cannot completely dispose it, as the limit of the latter is smaller. For this method the constraining limit can be obtained by finding

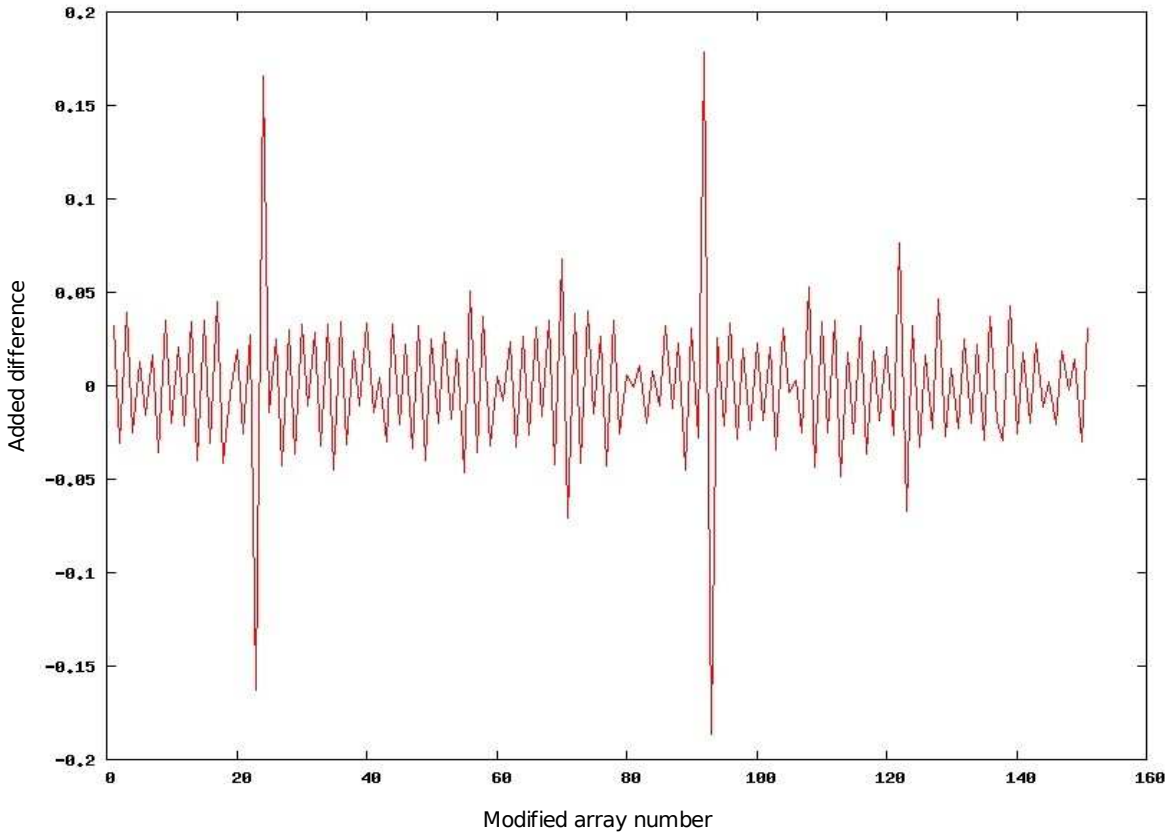

Fig 6-4: A plot of added differences for the previous spectrum. The improvement in the detection criterion is clearly apparent. The interference infected channels now have much larger deviations compared to the difference method.

the added maximum difference in the minimum standard deviation block and using the ratio of expected difference maxima for the entire spectrum(N-channels) to the block number(n_{blk} channels). It can also be calculated directly using (6.6). A similar limit as was estimated in the earlier method can be obtained and used in the interference detection. Since the probability of occurence of more than two subsequent differences of the same sign is quiet small, this ratio is slightly greater than that of the difference maxima. The approximate expected value of added difference maxima for N-channel gaussian noise is given by the empirical relation,

$$<Q_{mx}> \approx 2\sigma\, erf^{-1}\left(1 - \frac{0.5833}{2N}\right). \qquad (6.6)$$

Plots of expected absolute maximas obtained using randn function of octave to those given by the above expressions, (6.5) & (6.6) are given in the fig-5 for σ = 1. Again it is required to use a multiplicative factor(>1.0) with the value given by eqn-(6.6) .

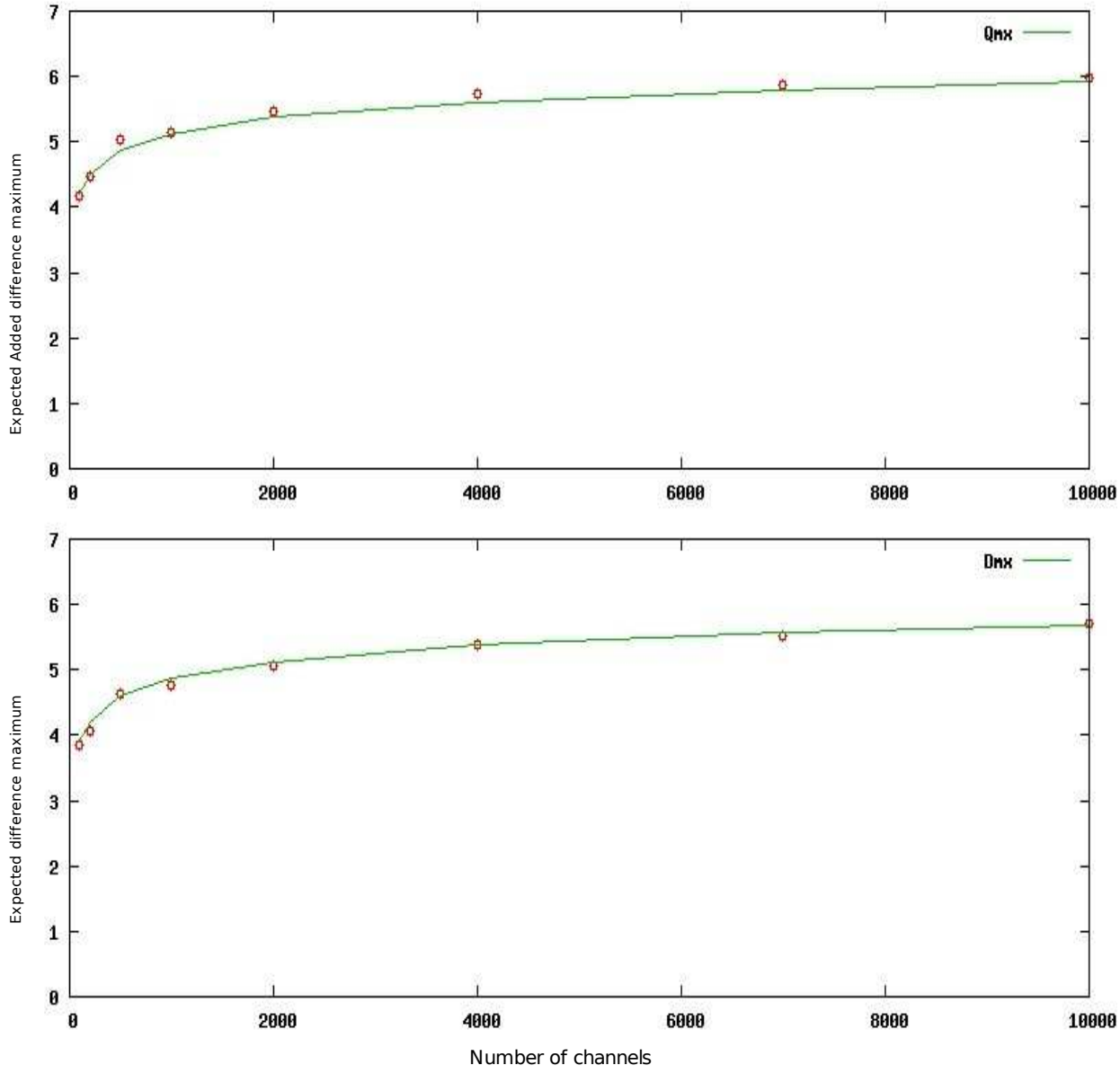

Fig 6-5: Comparision of D_{mx} and Q_{mx} between practically obtained values using octave's randn function and those obtained using equations (6) and (5) for σ = 1. The practical values are marked using circular dots while the solid line corresponds to calculated ones.

Appendix-II contains example codes which demonstrate the practical effectiveness of these methods. The following pages display their application to ORT data towards a few positions.

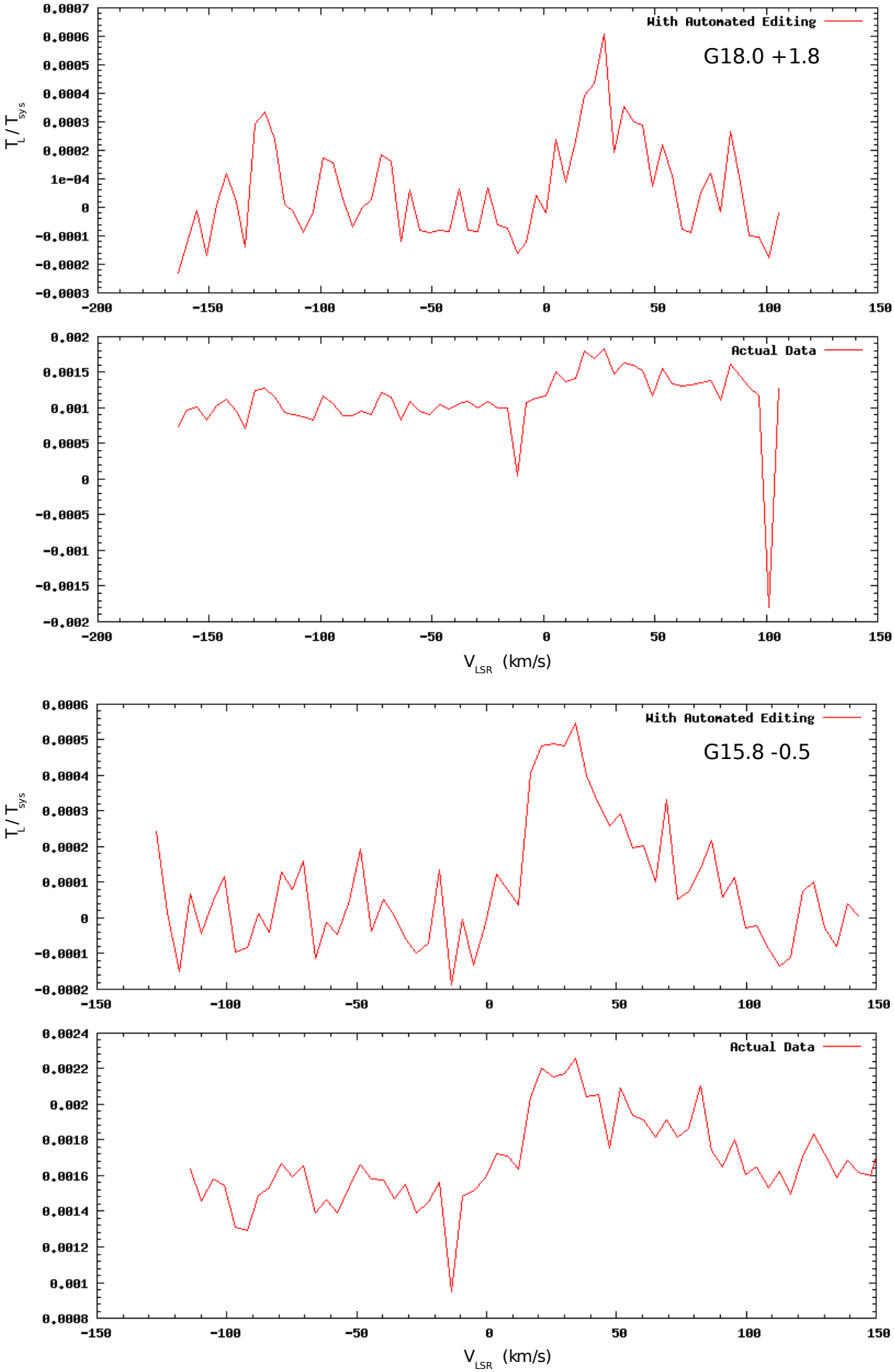

Fig 6-6: Examples of automated interference editing performed on ORT data as described in the text. The main features in the spectrum remain intact.

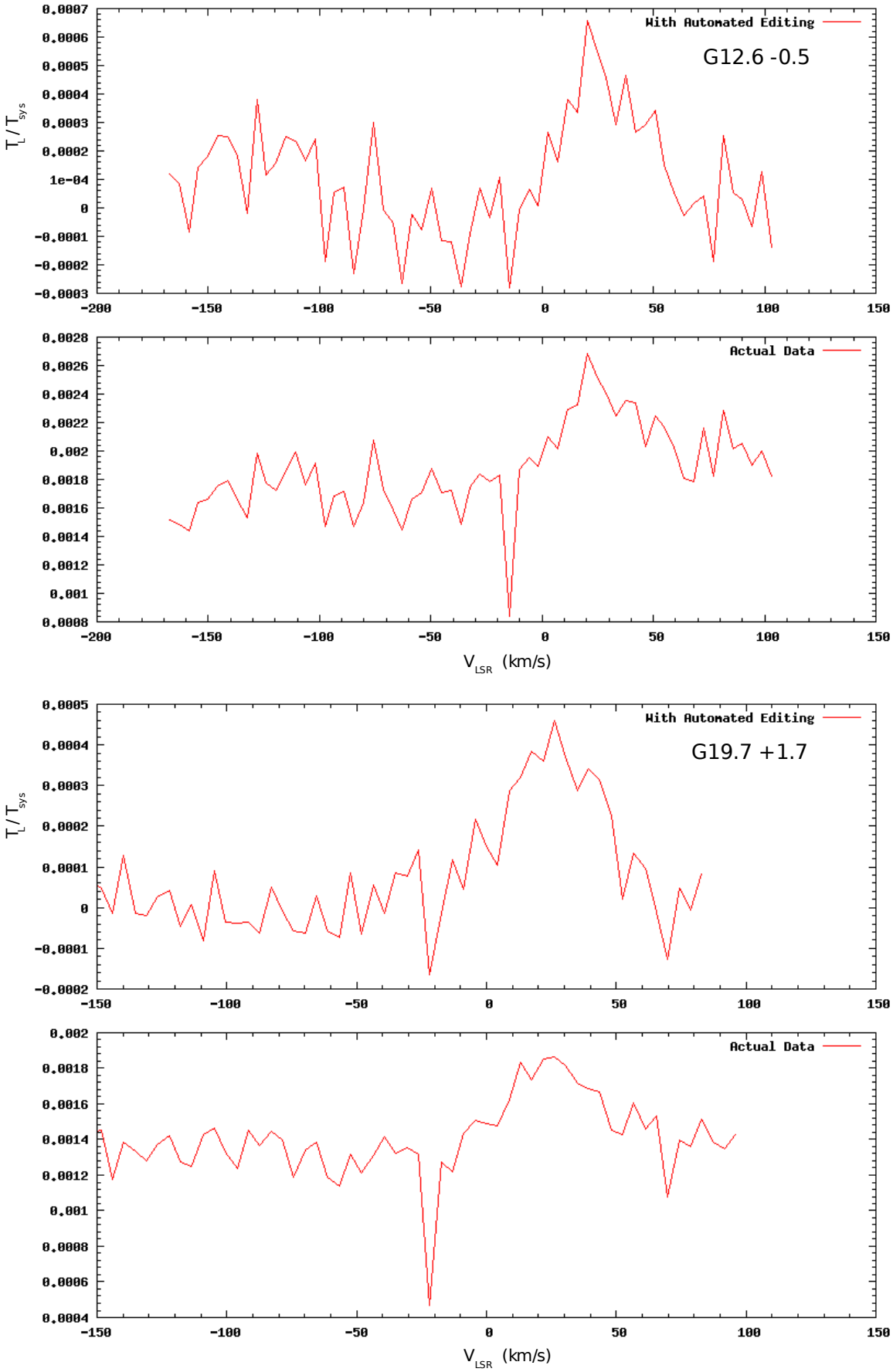

Fig 6-7: Examples of automated interference editing performed on ORT data as described in the text. The main features in the spectrum remain intact.

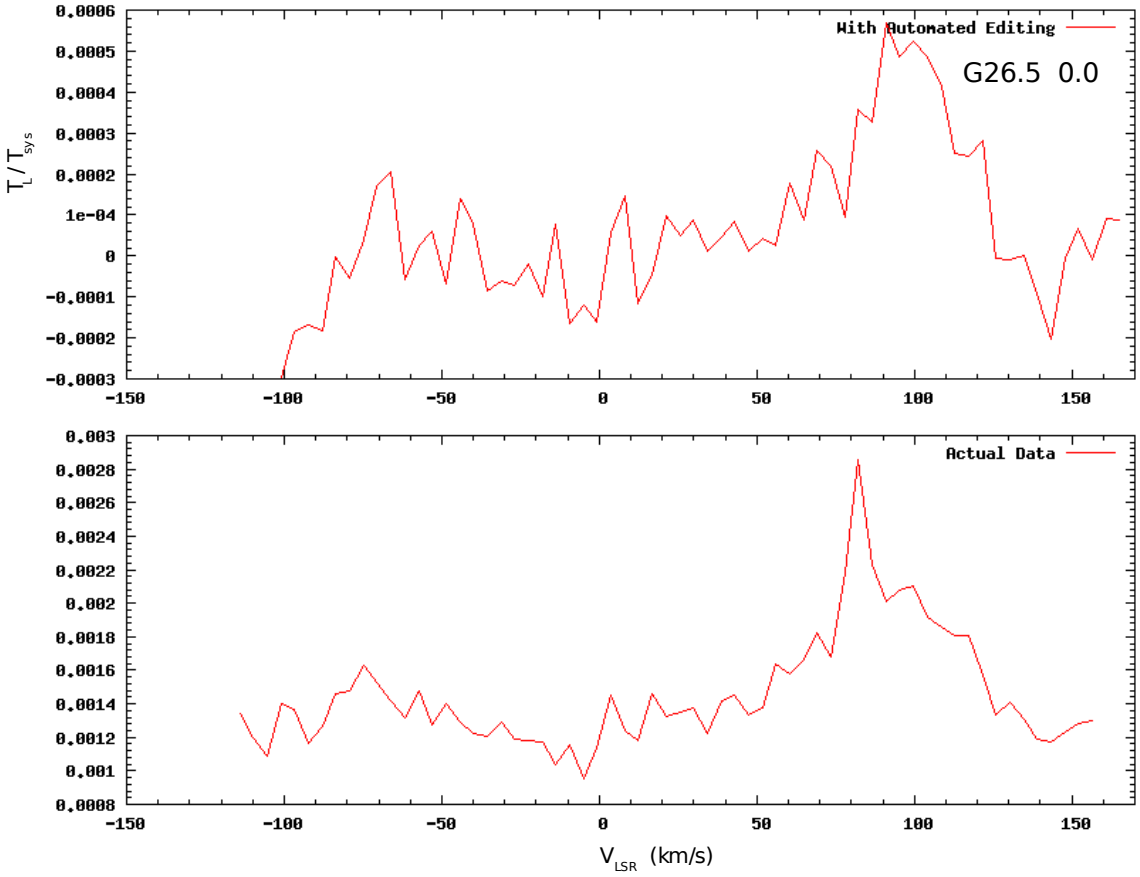

Fig 6-8: Examples of automated interference editing performed on ORT data as described in the text. The main features in the spectrum remain intact. This is good example, where the interference line is very close to the H271α . Features adjacent to the interference line are intact. It should be noted here that the automated editing has used less data as it discards some of it during processing, hence the difference in minor features.

APPENDIX-I

A An octave tool to edit interference manually from a spectrum

```
% ----------------          START OF PROGRAM          --------------------

% THIS IS AN OCTAVE CODE TO EDIT ANY NUMBER OF INTERFERENCE LINES
%  IN A SPECTRUM (1 – T_ON/T_OFF) OR AFTER THIS HAS BEEN FOLDED.

% The procedure for interference editing is as follows: When the program
% is run the raw spectrum with interference will be displayed. The user
% has to first click on the extreme left bottom corner and then on the
% extreme right top corner of the plot, this specific message will also be
% displayed on the screen. After this windows for interference editing will
% have to be specified by two clicks on either side of interference infected
% region. The code replaces the interference infected region with noise of
% standard deviation sampled from a region specified by the user. However
% the user can modify this according to his need. After clicking for
% interference editing, one has to specify a region for noise sampling by
% two clicks. The region between the clicks will be used for sampling the
% noise, this should be free of interference. It should be noted that the
% clicks are terminated by right clicks. After this the interference edited
% spectrum is displayed for baseline fitting. One can choose any number of
% portions of spectrum for baseline fitting by a series of clicks across the
% spectrum.

% -- Raju R Baddi

fid=fopen('input file','r'); % raw file (x,y) with interference.
fidw=fopen('outputfile','w'); % cleaned file (x,y') after editing.

grid on;
ld=N; % length of data or number of channels in the spectrum

for i=1:ld

        x(i)=fscanf(fid,"%f",1) ;
        y(i)=fscanf(fid,"%f",1) ;
        xref(i)=0;
        xindx(i)=i;

end

figure(1);

% setting the scales of abscissa and ordinate
% for plotting. 350 is for abscissa, 0.002 is for the
% ordinate. The user can change these as required.
```

```
axis([-350,300,-0.002,0.002]);

plot(x,y); hold on ; plot(x,xref); hold off;

no = input('Enter the number of editings u wish to make:');
disp('Select the y-axis values such that they lie on the baseline:');

[indx_x,indx_y] = ginput([-350 300 -0.002 0.002]);
z1=indx_x ;
z2=indx_y ;

disp('Click on the edge points of the region for estimating standard deviation \n ');

[sd_x,sd_y] = ginput([-350 300 -0.002 0.002]);
sd1=sd_x ;
sd2=sd_y ;
xrstd=xindx(x<sd1(2) & x>sd1(1));
sd=std(y(xrstd(1):xrstd(length(xrstd))));

for i=1:no

        xr=xindx(x<z1(2*i) & x>z1(2*(i-1)+1));


        m=(z2(2*i)-z2(2*(i-1) + 1))/(z1(2*i)-z1(2*(i-1) + 1))

        for j=xr(1):xr(length(xr))

                ytmp = y(xr(1)) + m*(j-xr(1))
                ygtmp = randn(1,1);

                y(j) = ytmp + ygtmp*sd ;

        end
end

figure(2); axis([-350 300 -0.002 0.002]) ;
plot(x,y);

%       BASELINE FITTING FOR THE SPECTRUM

figure(3);
axis([-350,300,-0.002,0.002]);
plot(x,y);hold on ; plot(x,xref); hold off;

[indx_x,indx_y] = ginput([-350 300 -0.002 0.002]);
z1=indx_x ;
z2=indx_y ;

no=length(z1);
no=no/2;
```

```
L=0;

for i=1:no

        xtmp=xindx(x<z1((i-1)*2+2) & x>z1((i-1)*2+1));
        x1(i)=xtmp(1);
        x2(i)=xtmp(length(xtmp));
        l=x2(i)-x1(i)+1;

        xp(L+1:L+l)=x(x1(i):x2(i)) ;
        yp(L+1:L+l)=y(x1(i):x2(i)) ;

        L=L+l;

end

p=polyfit(xp,yp,3); % fitting 3-order polynomial for baseline removal.

ypf=y.-(p(1)*x.^3 + p(2)*x.^2 + p(3)*x + p(4));

plot(x,y,x,ypf,'b');

fprintf(fidw,'%f  %f  \n', [x; ypf]);

fclose(fid); fclose(fidw);

%--------------------------     END OF PROGRAM   ------------------------
```

B A subroutine to resample a spectrum to a different resolution

---------- START OF SUBROUTINE ----------

```
% This subroutine resamples a given spectrum to a specified different resolution.
% The input variables are the reference abscissa to which the spectrum is to be
% resampled. The abscissa, the ordinate and the associated weights of the
% subject spectrum. The function returns the resampled spectrum along with
% the new weights. Here the new abscissa will be that of the reference spectrum.
% Name the subroutine as resmpsub.m

% -- Raju R Baddi

function [resmpld_y,wnew] = resmpsub(xref,x,y,w)

s = length(y)/2 + 1;
N = length(y);
jump = 1;
str_indx = 0;

vr1 = x(1)-x(2);
vr2 = xref(1)-xref(2);

dum = xref(1)-x(1);

if dum>0

        str_indx = round(dum/vr2);
        disp(str_indx);

        for i = 1:length(xref)-str_indx
         xrefn(i) = xref(str_indx+i);
        end

else
        jump = round(-1*dum/vr1);

        for i = 1:length(xref)
         xrefn(i) = xref(i);
        end

end

% for LOOP TO OBTAIN THE FOURIER TRANSFORM

for o = 1:N
 for q = 1:N
        b(q,o) = y(q)*exp(-sqrt(-1)*2*pi*(o-1)*(q-1)*1/N);
 end
```

```
end

YF = sum(b);

for i = 1:length(xref)-str_indx

        dum1 = floor(vr2*(i-1)/vr1) + jump + 1;

        if dum1==length(x)
         break;
        end

        comp1 = x(dum1)-xrefn(i);
        comp2 = x(dum1+1)-xrefn(i);

        if comp1<0
         comp1 = -1*comp1;
        else
         comp1 = comp1;
        end

        if comp2<0
         comp2 = -1*comp2;
        else
         comp2 = comp2;
        end

        if comp1<comp2
         shift = (x(dum1)-xrefn(i))/vr1;
         k = dum1;
        else
         shift = (x(dum1+1)-xrefn(i))/vr1;
         k = dum1+1;
        end

        wnew(i) = w(k);

        for l = 1:N
                a(l,dum1) = (1/N)*YF(l);
                a(l,dum1) = a(l,dum1)*exp(sqrt(-1)*2*pi*(k-1)*(l-1)*(1/N));

                if l<=s
                 ph(l) = exp(sqrt(-1)*2*pi*(l-1)*shift*(1/N));
                else
                 ph(l) = exp(sqrt(-1)*2*pi*(-2*s+l+1)*shift*(1/N));
                end

                a(l,dum1) = a(l,dum1)*ph(l);

        end

        p = sum(a);
```

```
        AF(i) = p(dum1);

end

resmpld_y = real(AF);
---------------------          END OF SUBROUTINE          ----------------------
```

C Subroutine to shift a spectrum by arbitrary channels(fractional part is also allowed)

---------- START OF SUBROUTINE ----------

```
% This code provides the shifting of a spectrum by arbitrary amount along
% the abscissa. This function is required if one were to average two spectra
% with different resolutions and with spectral lines at different abscissa(Vlsr)

% - Raju R Baddi

function [xshf,yshf,wshf]=specshift(x,k,y,Xrefc,Xc,w,xlm1,xlm2);

ldx=length(x);
xtmp=x; ytmp=y; ktmp=k; wtmp=w;

for i=1:ldx

        x(ldx-i+1)=xtmp(i);
        k(ldx-i+1)=ktmp(i);
        y(ldx-i+1)=ytmp(i);
        w(ldx-i+1)=wtmp(i);

end


ldxref=length(x);

vres=x(1)-x(2);

kc=k(1);

for i = 1:ceil((100-1)*(k(1)-k(2))/vres)
        kn(i) = kc - (i-1)*vres ;
        w(i)=1;
end

% xlm1 is the +ve limit for the new abscissa(here Vlsr).
% xlm2 is the -ve limit for the new abscissa(here Vlsr).

nx1 = fix((xlm1-x(1))/vres);
nx2 = fix((xlm2+x(length(x)))/vres);
n1 = fix((xlm1-kn(1))/vres);
n2 = fix((xlm2+kn(length(kn)))/vres);

dumyx1 = zeros(1,nx1);
dumyx2 = zeros(1,nx2);
dumy1 = zeros(1,n1);
dumy2 = zeros(1,n2);
```

```
% resampling subroutine
[yresmp,wnew] = resmpsub(kn,k,y,w);

% The display of resampled spectrum can be seen by uncommenting the next line

plot(k,y,'m'); hold on; plot(kn(1:length(yresmp)),yresmp,'g'); hold off;

y_extd = [dumy1,yresmp,dumy2];
wnew = [dumy1,wnew,dumy2];

for i = 1:nx1
        dumxx1(nx1+1-i) = x(1)+i*vres;
end

for i = 1:nx2
        dumxx2(i) = x(length(x))-i*vres;
end

for i = 1:n1
         dumx1(n1+1-i) = kn(1)+i*vres;
end

for i = 1:n2
        dumx2(i) = kn(length(kn))-i*vres;
end

xnewh = [dumxx1,x,dumxx2];

xnew = [dumx1,kn,dumx2];

[yshf,wshf] = specshift1(xnew,xnewh,y_extd,Xc,Xrefc,wnew);

xshf=xnewh;


----------------------  END OF SUBROUTINE  ----------------------------
```

C. 1 The subroutine specshift1

```
--------------------          START OF SUBROUTINE          ------------------

function [yf,wf] = specshift1(x1,x2,y1,xc1,xc2,w)

a = [];b = [];yf = [];wf = [];
s = length(y1)/2 + 1;
N = length(y1);
%disp(N);
vr1 = x1(1)-x1(2);
```

```
indx1 = floor((x1(1)-xc1)/(x1(1)-x1(2))) + 1;

indx2 = floor((x2(1)-xc2)/(x2(1)-x2(2))) + 1;

diff1 = x1(indx1)-xc1;
diff2 = x2(indx2)-xc2;

if diff1<0
        diff1 = -1*diff1;
end

if diff2<0
        diff2 = -1*diff2;
end

shift = (diff1-diff2)/vr1;

Y1F = fft(y1);
w1F = fft(w);

for k = 1:N

        for l = 1:N
                by(l,k) = (1/N)*Y1F(l)*exp(sqrt(-1)*2*pi*(k-1)*(l-1)*(1/N));
                bw(l,k) = (1/N)*w1F(l)*exp(sqrt(-1)*2*pi*(k-1)*(l-1)*(1/N));

                if l<=s
                        ph(l) = exp(sqrt(-1)*2*pi*(l-1)*shift*(1/N));
                else
                        ph(l) = exp(sqrt(-1)*2*pi*(-2*s+1+l)*shift*(1/N));
                end

                by(l,k) = by(l,k)*ph(l);
                bw(l,k) = bw(l,k)*ph(l);
        end

end

cy = real(sum(by));
cw = real(sum(bw));

for i = 1:length(y1)
         y1(i) = cy(i);
        w(i) = cw(i);
end

indxdiff = indx1-indx2

if indxdiff >= 0

        for i = 1:length(y1)-indxdiff
```

```
            y1new(i) = y1(i+indxdiff);
            w1new(i) = w(i+indxdiff);

        end

        yf = [y1new,zeros(1,length(x2)-length(y1new))];
        wf = [w1new,zeros(1,length(x2)-length(y1new))];

else

        indxdiff = -1*indxdiff;


        for i = 1:(length(x2)-indxdiff)
            yf(i+indxdiff) = y1(i);
            wf(i+indxdiff) = w(i);
        end

end

------------------------------  END OF SUBROUTINE  ----------------------
```

D Practical application of resampling and shifting Subroutines

A spectrum that was shifted using the codes given here has been shown on the next page. This was done to detect helium line at ~327MHz by averaging spectra towards different positions. Since the hydrogen line towards different positions would occur at different V_{LSR} the associated helium line would as well shift its position similarly. This required that all the spectra be aligned with respect to the V_{LSR} for their averaging. Also the spectra available were at different resolution due to different FFT point operation performed on the data. These codes provided the necessary tool to tackle this problem. The shift could be arbitrary since the lines may appear at arbitrary V_{LSR}. A reference spectrum was choosen and all the spectra were resampled and shifted to align with this having eventually a common resolution. Another example of shifting of spectra relevant to the current work with WSRT data is shown below.

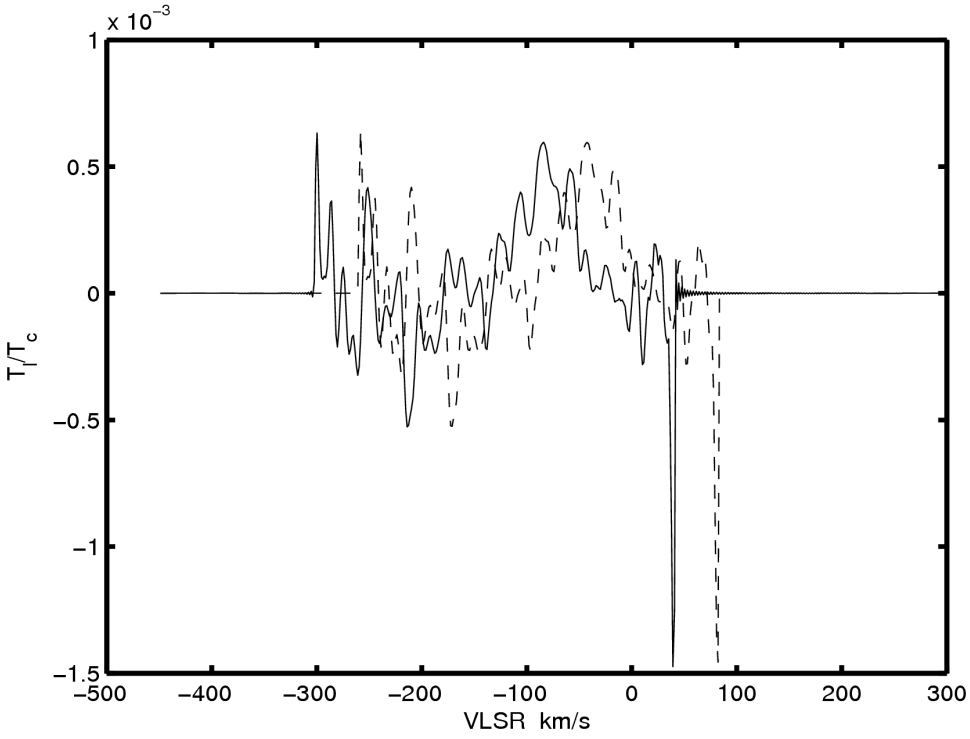

Fig AI-1: A fractional Fourier shifted spectrum from previous ORT observations used to detect the helium line(sec-5.6), using the programs given here. Note the features in the two spectra remain nearly the same. The bold spectrum(with ringing at the right end) is the shifted spectrum.

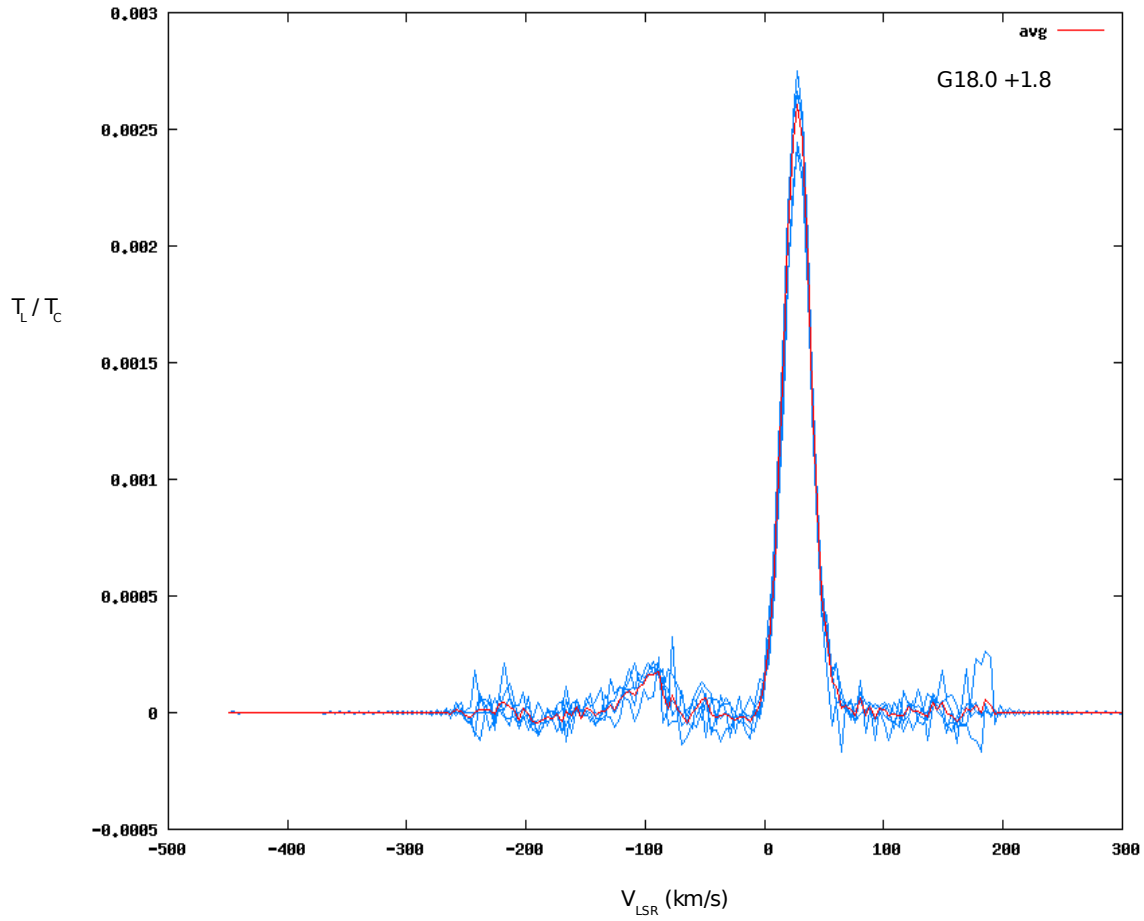

Fig AI-2: The fourier resampled & shifted spectra towards the position G18.0+1.8. The average of all the spectra is also shown in bold(red). The blue lines correspond to the individual shifted spectra corresponding to 1-2-4-5-6-7 bands, though one cannot make out these individually. The abscissa is V_{LSR} and the ordinate is T_L / T_C .

E Code to find the resulting RRL for a specified atom and other parameters.

```
function u=freq(n,s,se,Z)

        % description of arguments
        % n – principle quantum number
        % s – Δn , step involved in the transition
        % se – atom, 'h','he' or 'c'
        % Z – Charge on the central core

        % Raju R. Baddi

        %defining the constants
        u=1.66053886e-27; %atomic mass unit amu
        MHe=4.00260325415*u; %mass of the helium atom
        MH=1.00782503207*u;
        MC=12.0*u;
        Me=9.1093826e-31;
        h=6.6260693e-34;
        e=1.60217653e-19;
        Eo=8.854187817e-12;

        %defining the mass of single ionised helium nucleus
        MH_N=MH-Z*Me;
        MHe_N=MHe-Z*Me;
        MC_N=MC-Z*Me;

        if se=='h'

         mu=(Me*MH_N)/(Me+MH_N);
         K=constH=Z*Z*(e^4)*mu/(8.0*Eo*Eo*h*h*h);

        elseif se=='he'

        mu=(Me*MHe_N)/(Me+MHe_N);
        K=Z*Z*(e^4)*mu/(8.0*Eo*Eo*h*h*h);

        elseif se=='c'

         mu=(Me*MC_N)/(Me+MC_N);
         K=Z*Z*(e^4)*mu/(8.0*Eo*Eo*h*h*h);

        end

        u=K*(1./(n.^2) – 1./((n+s).^2));

%--------------        END OF PROGRAM          --------------------
```

F A code to search for probable transitions given the frequency of RRL and tolerance around it.

```
function [M_qn,Z_qn,qn,del_qn,nuf] = searchfreq(nu, delnu)

    % This function finds the possible transitions
    % that can result in the specified radio frequency, ν(nu)
    % within ±Δν (delnu)from ν.

    % Raju R. Baddi

    %ALL THE CALCULATIONS DONE IN SI UNITS

    % Description of the arguments
    % nu - sought after frequency
    % delnu - tolerance bandwidth across nu

    % M_qn – Atom number in the given list
    % Z_qn – charge on the core
    % qn – principal qauntum number
    % del_qn – steps involved in the transition
    % nuf – resulting frequency
    %-----------------------------------------------------
    % Defining the constants

    u=1.66053886e-27; %atomic mass unit amu, Kg
    Me=9.1093826e-31; %mass of the electron, Kg
    h=6.6260693e-34; %planck's constant, J/s
    e=1.60217653e-19; %charge of electron, C
    Eo=8.854187817e-12; %electrical permetivity,  F/m

    %-----------------------------------------------------
    % Write the mass of the atoms here

    MH=1.00782503207*u;  %mass of the hydrogen atom
    MHe=4.00260325415*u; %mass of the helium atom
    MC=12.0*u;           %mass of the carbon atom
    %-----------------------------------------------------


    %-----------------------------------------------------
    % List your atoms here

    MX = [MH, MHe, MC];
    Z = [1,2,6];  % number of ionization possible
    %-----------------------------------------------------

    m=1;
    for i=1:length(MX) % Type of atom H,He,C ...
```

```
for j=1:Z(i) % Number of ionizations for the same atom.

        MN = MX(i) - j*Me ; % Mass of core, nucleus + residual electrons.
        mu = (Me*MN)/(Me+MN) ; % μ is the reduced mass.

        Knu = j*j*(e^4)*mu/(8.0*Eo*Eo*h*h*h);


        for k=1:10  % Quantum step Δn involved in the transition.

                n = floor( ((2*k*Knu)/nu)^(1/3) );

                for p=1:11
                        nu_f = Knu*(1/(n-6+p)^2 - 1/(n-6+p+k)^2);

                        if abs(nu_f – nu)<delnu

                                qn(m) = n-6+p;
                                del_qn(m) = k;
                                 Z_qn(m) = j;
                                M_qn(m) = i;
                                nuf(i)=nu_f;
                                m=m+1;

                        end

                end
        end

 end

end
printf('\n');
printf('Atom - atom number in the list.\n');
printf('Z - is the charge on the core.\n');
printf('Other terms have usual meaning.\n');
printf('-------------------------\n');
printf('Atom  Z   n    Δn    ν(MHz)\n');
printf('-------------------------\n');

for i=1:m-1

        printf('%3d                %2d                %3d                %3d
%4.3f\n',M_qn(i),Z_qn(i),qn(i),del_qn(i),nuf(i)/1e6);

end

printf('-------------------------\n');


%--------------        END OF PROGRAM          --------------------
```

APPENDIX-II

A Derivation of expected maximum for N-channel gaussian noise

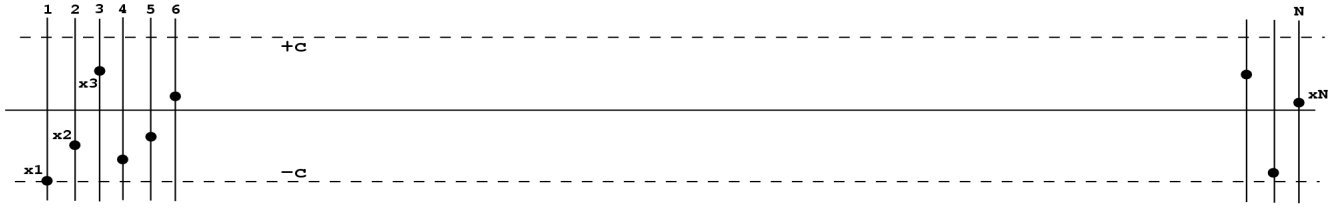

Fig 1: Calculating the expected maximum for a N-channel gaussian noise. The first channel is fixed at a value of -c. The other channels are allowed to take any value between +c and -c.

Considering N channels which produce gaussian noise of the same standard deviation σ. The probability that an absolute maximum value of c is produced in these channels is

$$p(c) = \frac{2^N N e^{\frac{-c^2}{2\sigma^2}}}{(\sigma\sqrt{2\pi})^N} \int_0^c e^{\frac{-x_2^2}{2\sigma^2}} dx_2 \int_0^c e^{\frac{-x_3^2}{2\sigma^2}} dx_3 \ldots \int_0^c e^{\frac{-x_N^2}{2\sigma^2}} dx_N \qquad \text{(AII-1)}$$

which is nothing but the integral over all the possibilities over the remaining N-1 channels keeping one of the channels fixed at c. The factor 2^N is due to the fact that both +ve & -ve values are possible in each channel. The remaining N-1 channels are allowed to take all the values between -c and +c. N in the numerator appears as there are N-ways to fix a channel value to c. Each of the above integrals evaluates to $\sigma\sqrt{\pi/2}\,erf(c/\sigma\sqrt{2})$ resulting in,

$$p(c) = \frac{2N e^{\frac{-c^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}} \left[erf\left(\frac{c}{\sigma\sqrt{2}}\right)\right]^{N-1} \qquad \text{(AII-2)}$$

Now the maximum expected value can be written as

$$< X_{mx} > = \int_0^\infty c\, p(c)\, dc \qquad \text{(AII-3)}$$

$$<X_{mx}> \;=\; \frac{2N}{\sigma\sqrt{2\pi}} \int_0^{\infty} c\, e^{\frac{-c^2}{2\sigma^2}} \left[erf\left(\frac{c}{\sigma\sqrt{2}} \right) \right]^{N-1} dc \qquad \text{(AII-4)}$$

which can also be written using the derivative of $S \;=\; \left[erf\left(\frac{c}{\sigma\sqrt{2}} \right) \right]^{N}$ and expressing c in terms of S, as

$$<X_{mx}> \;=\; \sigma\sqrt{2} \int_0^1 erf^{-1}(S^{1/N})\, dS \qquad \text{(AII-5)}$$

which has an approximation

$$<X_{mx}> \;\approx\; \sigma\sqrt{2}\;\; erf^{-1}\left[1 \;-\; \frac{0.5833}{N} \right] \qquad \text{(AII-6)}$$

which has been derived by considering the series expansion for $erf^{-1}(S)$.

$$\begin{aligned} \int_0^1 erf^{-1}(S^{1/N})\, dS \;&=\; \sum_0^{\infty} \frac{C_k}{2k+1} \left(\frac{\sqrt{\pi}}{2} \right)^{2k+1} \int_0^1 S^{\frac{2k+1}{N}}\, dS \\ &=\; \sum_0^{\infty} \frac{C_k}{2k+1} \left(\frac{\sqrt{\pi}}{2} \right)^{2k+1} \left[\frac{S^{\frac{2k+N+1}{N}}}{\frac{2k+N+1}{N}} \right]_0^1 \\ &=\; \sum_0^{\infty} \frac{C_k}{2k+1} \left(\frac{\sqrt{\pi}}{2} \right)^{2k+1} \frac{N}{2k+N+1} \end{aligned}$$

Considering $erf^{-1}(z)$, such that,

$$\begin{aligned} z^{2k+1} \;&=\; \frac{N}{2k+N+1} \\ &=\; \left[1+\frac{2k+1}{N} \right]^{\frac{-1}{2k+1}} \end{aligned} \qquad \text{(AII-7)}$$

Since N is normally large and the higher(k) order terms in the series contribute less and less. The above equation has a simple approximation,

$$z = 1 - \frac{1}{N} \tag{AII-8}$$

A better approximation is obtained by binomially expanding(7) and considering the first 5 terms. This introduces +ve errors in the lower order terms but helps to account for higher order terms. In this approximation 2k+1 ~ N for all the terms. This yields the approximation,

$$z = 1 - \frac{0.5833}{N} \tag{AII-9}$$

and hence (6).

B Derivation of expected maximum difference for N channel gaussian noise

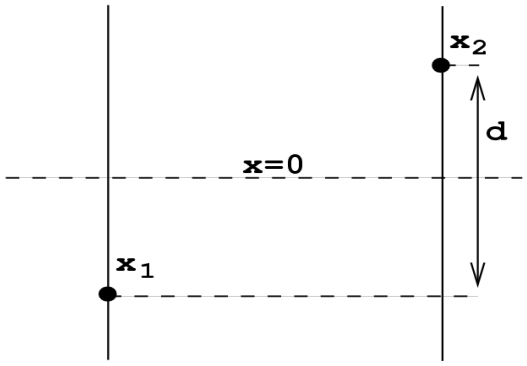

Fig 2: Probability of occurence of d in 2-channel gaussian noise.

To get the maximum expected difference between two adjacent channels in a N channel gaussian noise spectrum, first the probability of occurence of a difference of d in two channels is calculated. Further it is assumed that each of the adjacent channels behave approximately independently and the two channel probability distribution is applied to the N channels. This approximation becomes better when one considers a large number of channels as can be seen by the analysis of a 3 channel gaussian noise. Refering fig-2 one can write for the probability of occurence of a difference d in a 2-channel gaussian noise as,

$$p_2(d) = \frac{1}{2\pi\sigma^2}\int_{-\infty}^{\infty} e^{-\frac{x^2+(x+d)^2}{2\sigma^2}}\,dx \qquad \text{(AII-10)}$$

so the probability density for a 2-channel difference is,

$$p_2(d) = \frac{1}{2\sigma\sqrt{\pi}}\, e^{-\frac{d^2}{4\sigma^2}} \qquad \text{(AII-11)}$$

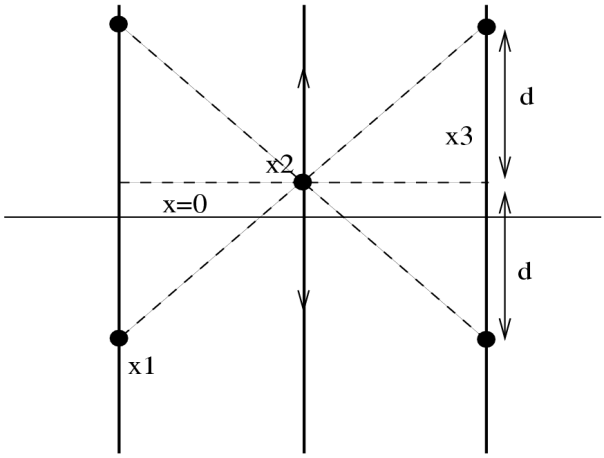

Fig 3: Analysis of a 3-channel gaussian noise for difference maxima between adjacent channels.

which is similar to gaussian noise with standard deviation $\sigma\sqrt{2}$. Now the 3-channel analysis for adjacent differences is as follows, refering fig-3 x1 & x2 are fixed as such that the difference between them is d. x3 is allowed to take any value such that the its magnitude of difference with x2 does not exceed d. Next the same is repeated to account the other possibility of x2 and x3 being fixed and x1 being free to take variable values. The net probability would be integrating over x2 from -∞ to ∞ . So we can write this net probability density as,

$$p_{3_2}(d) = \frac{2\cdot 2}{\left(\sigma\sqrt{2\pi}\right)^3}\int_{-\infty}^{+\infty} e^{-\frac{x^2+(x+d)^2}{2\sigma^2}}\int_{-d}^{+d} e^{\frac{-(x+y)^2}{2\sigma^2}}\,dy\;\;dx \qquad \text{(AII-12)}$$

$$= \frac{1}{\pi\sigma^2}\int_{-\infty}^{+\infty} e^{-\frac{x^2+(x+d)^2}{2\sigma^2}}\left(erf\left(\frac{d+x}{\sigma\sqrt{2}}\right)+erf\left(\frac{d-x}{\sigma\sqrt{2}}\right)\right)dx \qquad \text{(AII-13)}$$

A factor of 2 in the numerator is due to the fact that d in the first integrand takes 2 values +ve and -ve and in both the cases it integrates to the same amount. Another 2 is due to the interchange of roles between x1 and x3. Now when a similar analysis is done by considering 4 channels and taking the difference in two channels independently, we see that the probabilities are approximately the same for occurence of a difference d in the two cases. The probability in the case of 4 channels, but pair wise is

$$\begin{aligned} p_{4_2}(d) &= \frac{2}{\pi\sigma^2}\int_{-\infty}^{+\infty} e^{-\frac{x^2+(x+d)^2}{2\sigma^2}} \int_{-d}^{+d} p_2(y)dy \;\; dx \\ &= \frac{2}{\sigma\sqrt{\pi}} e^{-\frac{d^2}{4\sigma^2}} erf\left(\frac{d}{2\sigma}\right) \end{aligned} \qquad \text{(AII-14)}$$

This calculation can also be done by applying $p_2(d)$ to both the pairs. It can be checked that for a given value of d,

$$p_{3_2} \approx p_{4_2} \qquad \text{(AII-15)}$$

Hence one can assume that the channels behave approximately independently pair wise in a series of N channels. By considering the difference probability distribution for 2 channels and treating the N-1 channel pairs as new single channels in which the occurence value is not 'x' but 'd' we can write using the same derivation as for X_{mx} for the maximum expected difference value in a N channel gaussian noise as,

$$< D_{mx} > \approx \frac{N-1}{\sigma\sqrt{\pi}} \int_0^{\infty} d\, e^{-\frac{d^2}{4\sigma^2}} \left[erf\left(\frac{d}{2\sigma}\right) \right]^{N-2} dx \qquad \text{(AII-16)}$$

which again using $S = \left[erf\left(\frac{d}{2\sigma}\right) \right]^{N-1}$ can be written as

$$< D_{mx} > = 2\sigma \int_0^1 erf^{-1}(S^{1/(N-1)})\, dS \qquad \text{(AII-17)}$$

For large N the -1 in the exponent can be dropped and simply one can use

$$< D_{mx} > \; = \; \sqrt{2} < X_{mx} > \qquad \text{(AII-18)}$$

A plot of comparision of calculated maxima values using above analysis and those obtained practically using randn function in octave are displayed in fig-5. Maxima and difference maxima for different channel numbers are found using randn function. An average of every 5 trials is recorded. Such sets of 5 trials are repeated 50 times. These 50 values are compared by plotting with those given by expressions (6) & (18). These are the straight lines in the plots.

The following pages contain octave test codes which return the flags for blocks of data infected with interference using the technique described above. However it is easy to trace to the exact infected channels.

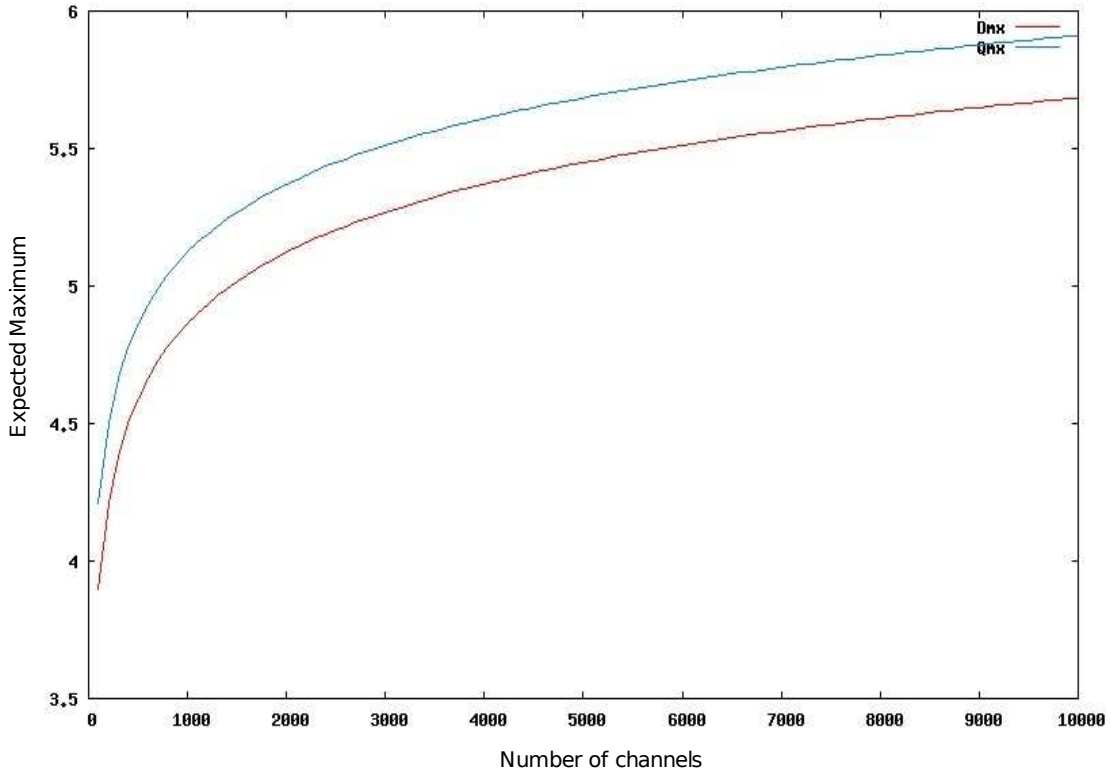

Fig 4: Plot of Q_{mx}(added difference maxima) and D_{mx}(difference maxima) Vs N, for N=100 to 10000 with $\sigma = 1$.

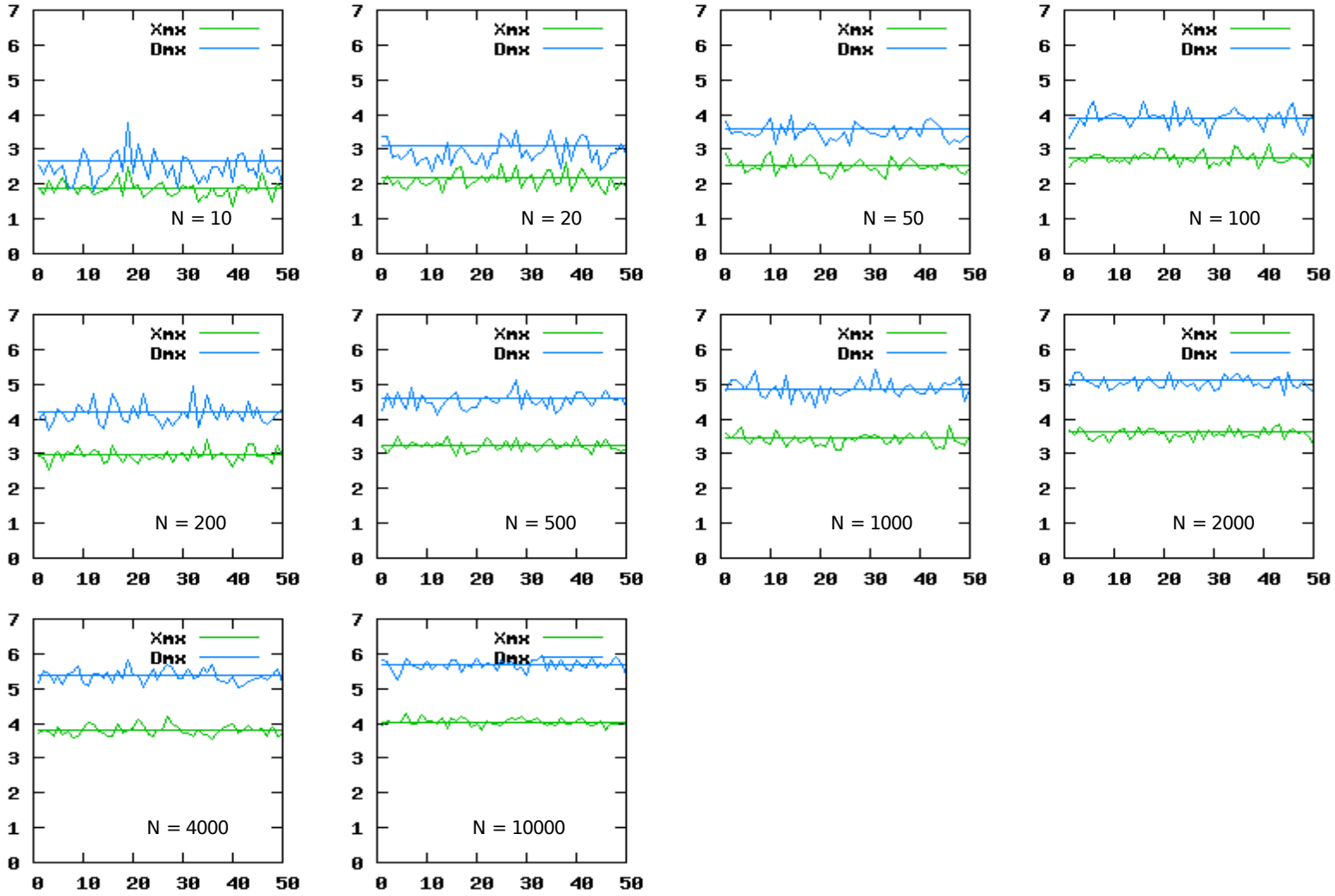

Fig 5: Comparing the expected maxima and difference maxima with calculated values for σ = 1 . The straight line is that calculated theoretically while those obtained from octave('randn' function) are the scattered values. Each value is an average of 5 maxima samples. The abscissa is the trial number.

P1 Subroutine to demonstrate interference detection using expected maxima in gaussian noise of N-channels.

```
------------          START OF SUBROUTINE          --------------
% Here the spectrum is divided into n sub blocks which is decided
% by the user specified block size(blk_size). In each block
% standard deviation(sd) is found. The block with minimum sd
% is used as a reference block to sample the noise in data. This
% sd is used to calculate expected maxima values in gaussian
% noise across the spectrum.  This method is useful only when
% the spectrum has more or less zero mean. If this is not so
% it is better to omit this method.  Further this is just a test code
% and has no practical value. It is just to demonstrate the feasibility
% of this method.

% - Raju R Baddi

function intfr_blks=intfr_det_DDmthd(data,blk_size)

l_d=length(data);
no_blks=floor(l_d/blk_size);
d1=data;

for i=1:no_blks
        Dmax(i)=max(abs(d1((i-1)*blk_size+1:i*blk_size)));
        sd(i)=std(d1((i-1)*blk_size+1:i*blk_size));
end

range=Dmax;
[sd_mn,sd_mni]=min(sd)
min_range=range(sd_mni);

for i=1:no_blks

        if range(i) > sd_mn*sqrt(2)*erfinv(1 – 0.5833/ld)
                intfr_blks(i)=1;
        else
                 intfr_blks(i)=0;
        end

end

--------------          END OF SUBROUTINE          --------------
```

P2 Subroutine to demonstrate interference detection using expected difference maxima in gaussian noise of N-channels.

```
------------          START OF SUBROUTINE          -------------
% Here the spectrum is divided into n sub blocks which is decided
% by the user specified block size(blk_size). In each block
% standard deviation(sd) is found. The block with minimum sd
% is used as a reference block to sample the noise in data. This
% sd is used to calculate expected difference maxima values in gaussian
% noise across the spectrum.  Further this is just a test code  and
% has no practical value. It is just to demonstrate the feasibility
% of this method.

% - Raju R Baddi

function intfr_blks=intfr_det_Dmthd(data,blk_size)

l_d=length(data);
no_blks=floor(l_d/blk_size);

for i=1:l_d-1
        d1(i)=data(i+1)-data(i);
end

for i=1:no_blks
        Dmax(i)=max(abs(d1((i-1)*blk_size+1:i*blk_size)));
        sd(i)=std(d1((i-1)*blk_size+1:i*blk_size));
end

range=Dmax;
[sd_mn,sd_mni]=min(sd);
min_range=range(sd_mni);

xlabel(' Channel Number ');
ylabel(' Adjacent Channel Differences ');
plot(d1); pause;

for i=1:no_blks
        if range(i) > 2*sd_mn*erfinv(1 – 0.5833/ld)
                intfr_blks(i)=1;
        else
                intfr_blks(i)=0;
        end
end

-----------          END OF SUBROUTINE          ---------------
```

P3 Subroutine to demonstrate interference detection using expected added difference maxima in a N-channel gaussian noise.

```
----------        START  OF  SUBROUTINE        ----------
% Here the spectrum is divided into n sub blocks which is decided
% by the user specified block size(blk_size). In each block
% standard deviation(sd) is found. The block with minimum sd
% is used as a reference block to sample the noise in data. This
% sd is used to calculate expected added difference maxima
% values in gaussian noise across the spectrum.  Further this
% is just a test code  and has no practical value. It is just to
% demonstrate the feasibility  of this method.  However portions
% of the code can be used to build a real practically useful
% program.

% - Raju R Baddi
function intfr_blks=intfr_det_ADmthd(data,blk_size)

l_d=length(data);
no_blks=floor(l_d/blk_size);

for i=1:l_d-1
        d1(i)=data(i+1)-data(i);
end

cntr=1;
for k=1:no_blks
        sd(k)=std(data(1+(k-1)*blk_size:k*blk_size));

        i=1+(k-1)*blk_size;seed=0;j=1;sum_d1=[];
        while i<blk_size*k

                if d1(i)>0
                        while d1(i)>0
                                seed=seed+d1(i);
                                i=i+1;
                        end
                else
                        while d1(i)<=0
                                seed=seed+d1(i);
                                i=i+1;
                        end
                end

        sum_d1(j)=seed;
        ADarry(cntr)=seed; cntr=cntr+1; seed=0;
```

```
        j=j+1;

        end

        ADmax=max(abs(sum_d1));
        range(k)=ADmax;

end

[sd_mn,sd_mni]=min(sd);
min_range=range(sd_mni);

xlabel(' Modified Abscissa ');
ylabel(' Added Channel Differences ');
plot(ADarry);

for i=1:no_blks

        if range(i)>2*sd_mn*erfinv(1 – 0.5833/(2*ld))
                intfr_blks(i)=1;
        else
                intfr_blks(i)=0;
        end

end

----------        END  OF  SUBROUTINE          ----------
```

It should be noted that the range limit ("range(i) > " statement) should be mulitplied by a muliplicative factor(>1.0, but not done in the above programs) as has been mentioned in the text.

References

Altenhoff, W.J., Mezger, P.G., Wendeker, H. and Westerhout, G., 1960, Veroffentlinchungen der Universitats Sternwarte Zu Bonn, No. 59, 48 .

Altenhoff, W.J., Downes, D., Pauls, T. and Schraml, J., 1979, *A&AS*...**35**...23A

Anatharamaiah, K.R., Radhakrishnan, V. and Shaver, P.A., 1984, *Astronomy and Astrophysics* **138**, 131A .

Anatharamaiah, K.R., Erickson, W.C., and Radhakrishnan, V., 1985, *Nature* **315**, 647.

Anish Roshi. D. ,1995, M.Sc Thesis, Pune University , Pune.

Beigman, I.I., 1977, *Zh. Ehksp. Teor. Fiz.* **73**, 1729. English translation: 1977 Sov. Phys.-JETP, 46, 908 .

Burton, W.B. And Gordon, M.A., 1978, *Astron. Astrophys.* **63**, 7 .

Dravskikh, A.F., Dravskikh, Z.V., Kolbasov, V.A., Misezhnikov, G.S., Nikulin, D.E. and Shteinshleiger, V.B., 1965, *Dok. Akad. Nauk SSSR* **163**, 332. English translation: 1996, Sov. Phys. - Dokl. 10, 627 .

Egorova, T.M. and Ryzkov, N.F., 1960, *Izv. Glavn. Astrofiz. Obs.* **21**, 140 .

Gee, C.S., Percival, I.C., Lodge, I.C. and Richards, D., 1976, *Mon. Not. R. Astron. Soc.* **175**, 209 .

Goldberg, L. and Dupree, A.K., 1967, *Nature* **215**, 41 .

Griem, H.R., 1960, *Astrophys. J.* **149**, 883 .

Griem, H.R., 1967, *Astrophys. J.* **148**, 547 .

Griem, H.R., 1974, *Spectral line broadening by plasmas*, Vol. 39 of Pure and applied physics, Academic Press, New York.

Haffner, et. Al, 2003, *ApJS*, **149**, 405 .

Heiles, C., 1994, *Astrophys. J.* **436**, 720 .

Heiles, C., Reach, W.T. and Koo, B.-C., 1996a, *Astrophys. J.* **466**, 191 .

Heiles, C., Koo, B.-C., Levenson, N.A. and Reach, W.T., 1996b, *Astrophys. J.* **462**, 326 .

Kardashev, N.S., 1959, *Astron. Zh.* **36**, 838. English translation: 1960 Sov. Astron. - AJ 3, 813 .

Kulkarni, S.R., Heiles, C., Interstellar processes; *Proceedings of the Symposium*, Grand Teton National Park, WY, July 1-7, 1986 (A88-14501 03-90). Dordrecht, D. Reidel Publishing Co., 1987, p. **87**-122 .

Lilley, A.E., Palmer, P., Penfield, H. and Zuckerman, B., 1966, *Nature* **211**, 174 .

Lockman, F.J., 1976, *Astrophys. J.* **209**, 429 .

Lockman, F.J., 1989, *Astrophys. J., Suppl. Ser.* **71**, 469 .

McKee, C.F. And Ostriker, J.P., 1977, *Astrophys. J.* **218**, 148 .

Menzel, D.H., 1937, *Astrophys. J.* **85**, 330 .

Mezger, P.G., 1978, *Astron. Astrophys.* **70**, 565 .

Murray, Norman., Rahman, Mubdi., 2010*ApJ*...**709**..424M

Oster, L., 1961, *Rev. Mod. Phys.* **33**, 525 .

Osterbrock, D.E., 1989, *Astrophysics of Gaseous Nebulae and Active Galaxies*, University

Science Books, Mill Valley, California .

Paladini R., Burigana C., Davies R. D., Maino D., Bersanelli M., Cappellini B., Platania P., Smoot G. , 2003, *Astron. Astrophys.* **397**, 213 .

Petuchowski, S.J. And Bennet, C.L., 1993, *Astrophys. J.* **405**, 591 .

Reich, W. , Furst, E., Reich, P. and Reif, K., 1990, *Astron Astrophys Suppl Ser.* **85**, 633

Roshi, D. A., Anantharamaiah, K. R., 2000, *Astrophys. J.* **535**, 231 .

Roshi, D.A., Anantharamaiah, K. R., 2001, *Astrophys. J.* **557**, 226 .

Shaver, P.A., 1975, *Pramana* **5**, 1 .

Shaver, P.A., 1976a, *Astron. Astrophys.* **49**, 149 .

Shaver, P.A., 1976b, *Astron. Astrophys.* **49**, 1 .

Smirnov, G.T., Sorochenko, R.L. and Pankonin, V., 1984, *Astron. Astrophys.* **135**, 116 .

Smirnov, G.T., 1985, PhD thesis, Lebedev Physical Institute, Moscow, USSR .

Sobelman, I.I., Vainshtein, L.A. And Yukov, E.A., 1995, *Excitation of Atoms and Braodening of Spectral Lines*, Vol. 15 of *Springer Series on Atomic, Optical and Plasma Physics*, second edn, Springer, New York .

Sorochenko, R.L. And Borodzich, E.V., 1966, *Trans. IAU* **XIIB**, 360. Paper presented by Vitkevich at the XII General Assembly of the IAU in Hamburg, Germany, 1964 .

Stromgren, B., 1939, *Astrophys. J.* **89**, 526 .

Swarup, G., Sarma, N.V.G., Joshi, M.N., Kapahi, V.K., Bagri, D.S., Damle, S.H., Ananthakrishnan, S., Balasubramanian, V., Bahve, S.S. And Sinha, R.P., 1971, *Nature*, Phys. Science (Lond), **230**, 185 .

Terzian, Y. and Parrish, A., 1970, *Astrophys. Letters* **5**, 261 .

T.Prabu et. al. 2010, Thesis

van de Hulst, H.C., 1945, *Nederladsch Tidjschrift voor Naturkunde* **11**, 230 .

G. Westerhout 1958, B.A.N. **14**, 215 .