Gravitational wave radiometer

Data analysis proposal to the stochastic source group from Al Lazzarini and Rai Weiss and hopefully others.

Introduction:

We suggest a data analysis procedure that measures the gravitational wave strain power spectrum from a specific location on the sky and references the spectrum to other points for control of systematic errors. Initially we are considering the galactic center and the Virgo cluster as specific sources to be measured with LLO and LHO to test the idea. Once the programs for single sources are working, we intend to extend the analysis to generate a gravitational wave source map of the sky with a pixelization appropriate to independent points on the sky for frequencies upto 1kHz. When VIRGO comes on the air we would like to extend the method to using VIRGO and LHO as well as VIRGO and LLO. The baselines will allow better sky resolution and the combination of signals from all three detectors should reduce the effective beam sidelobes and noise.

The basic concept:

The idea is an extension of the techniques now being used to set limits on an isotropic stochastic background and combines this technique with the correlation techniques being used in the unmodeled burst search. The idea is to “point” the pair of interferometers at different sites by delaying the signal from one by the time it takes the gravitational wave to propagate from one site to the other and then perform the cross correlation. The delay between the sites will be a function of the source position on the sky and will change throughout the sidereal day as the Earth rotates about its axis. If gravitational wave sources exist at the chosen location, the time integral of the cross correlated power from the sources will grow linearly in time while the integrated noise power only grows as the square root of the time. The ability to compare the cross correlation integrals for the chosen spot on the sky with other spots using different delays with equal integration times gives a powerful technique to estimate the systematic errors in the measurement. With this delayed cross correlation technique the two detectors are effectively brought to the same location and the frequency response of the combined detectors is the same as that of a single detector but only for signals incident from that specific point in the sky. The spectral filters used on the outputs of the individual detectors will still need to be optimized to minimize the random noise against a posited spectrum from the source. Furthermore, there is no change in the random noise by pointing the detector system.

The mathematics:

The formulation of the cross spectrum of two detectors in measuring an isotropic background of gravitational waves was worked out by Peter Michelson (1989) for bar detectors and by Nelson Christensen (1990,1992) for interferometers. Refinements on the formulation for interferometers were made by Eanna Flanagan (1993). A technique using an expansion in spherical harmonics modulated by the Earth’s rotation as a method to search for anisotropies in a gravitational wave background was published by Bruce Allen and Adrian Ottewill (1997). Bruce Allen and Joe Romano (1999) analysed the signal processing strategies for detecting an isotropic background with optimal filtering, describing the methods that are now being used by the stochastic background search with LIGO data. Most recently Neil Cornish has posted a paper on mapping the sky with LIGO and LISA (2001), the methods being proposed are similar (but not identical) to the Cornish paper which was discovered after we began posing the problem for ourselves.

Christensen (1990) shows that the cross correlation spectrum of the signals from two detectors, one at site 1 and the other at site 2, due to a gravitational wave source in the **k** direction is given by

$$P(\omega) = [(h_+(\omega))^2 F_{1+}F_{2+} + (h_x(\omega))^2 F_{1x}F_{2x}] \cos(\vec{k} \cdot \overrightarrow{(p_1 - \vec{p}_2)} - \omega\tau)$$

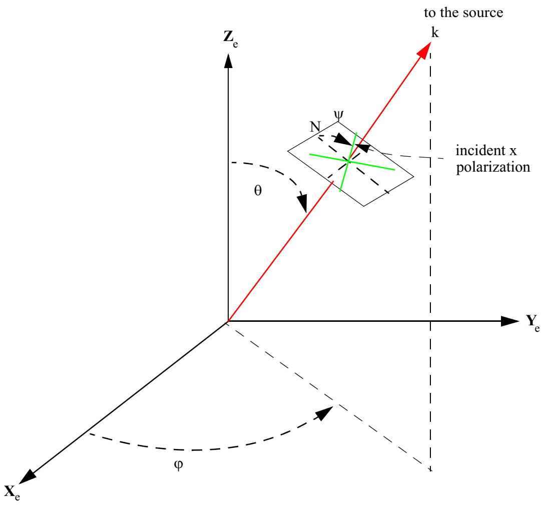


Figure 1 Source coordinates

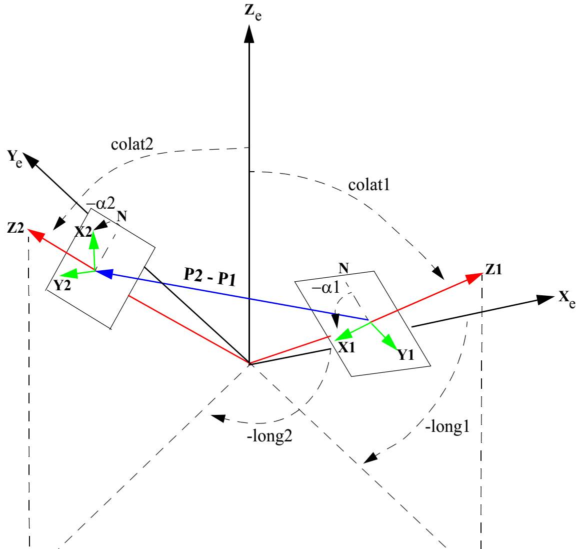


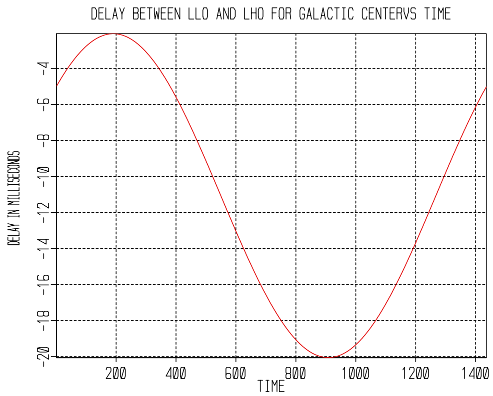
Figure 2 Detector coordinates

$h_+(\omega)$ and $h_x(\omega)$ are the amplitude spectra of the two gravitational wave polarizations coming from the source. **Figure 1** shows the coordinates for the source in the equatorial celestial coordinate system where φ is the hour angle and θ is the codeclination. The source is in the direction **k** with the usual magnitude ω/c where c is the velocity of light and ω the angular frequency of the gravitational wave. The polarization of the gravitational wave is designated by the angle ψ measured relative to the plane defined by k and the Z axis. The coordinate are tied to the inertial frame. The X axis lies along RA=0 and Z axis lies along the Earth’s rotation axis. **Figure 2** describes the detector coordinates. As the Earth turns the detector coordinate system rotates clockwise about the Z axis of the inertial frame of **Figure 1** once per sidereal day. The two detectors vertices are defined by their colatitude and longitude. (As an example, Site 1 is LLO and site 2 LHO.) The angle α is the azimuth angle of the x interferometer arm measured relative to the north direction in the detector plane. In the calculations, the x and y detector arms are assumed to be perpendicular. F_{1+} and F_{2+} are the angular response functions for + polarized waves at sites 1 and 2, while F_{1x} and F_{2x} are for the x polarization. The angular response functions are the well known ‘peanut” patterns determined from

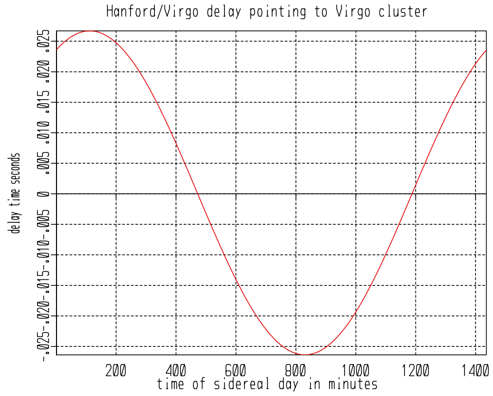
$$F_+ = \frac{1}{2}(1 + \cos^2(\Theta))\cos(2\Phi)\cos(2\Psi)\text{-}\cos(\Theta)\sin(2\Phi)\sin(2\Psi))$$

$$F_- = \frac{1}{2}(1 + \cos^2(\Theta))\cos(2\Phi)\sin(2\Psi)\text{-}\cos(\Theta)\sin(2\Phi)\cos(2\Psi))$$

The angle Θ extends between direction k toward the source and the detector z axis, while the angle Φ lies between the detector x arm and the plane containing the vector k and detector z axis. Finally, the angle Ψ is measured between the detector x arm and the gravitational wave polarization projected into the detector plane. All of these angles change during the sidereal day and are in good measure responsible for the complexity seen in the beam maps.



TIME DELAY BETWEEN LLO AND LHO VS MINUTES OF THE SIDEREAL DAY



IME DELAY BETWEEN LHO AND VIRGO VS MINUTES OF THE SIDEREAL DAY

Figure 3 Time delay between LLO and LHO

The relevant kernel of the expression is the argument of the cosine which does the beam pointing. **p1-p2** is a vector that links the vertices at the two sites with a length equal to the straight line distance between the two sites. The inner product of **k**/ω and **p1-p2** is the travel time of the gravitational wave between the sites. When the signal from one detector is delayed by a time τ relative to that from the other detector by the travel time, the argument of the cosine vanishes for all frequencies, and the combined detector sensitivity becomes frequency independent (except for the spectrum of the individual detectors). One is sitting on the zero order ("white light gravitational wave") interference between the two sites, or another way to think about it, one has brought the two detectors to the same place but only for that location on the sky. **Figure 3** shows the delays as a function of time in the sidereal day associated with looking at the Galactic center and the Virgo cluster with LLO and LHO. Around the pointing direction, the cosine oscillates. A single unipolar detection patch on the sky has the angular width given by $\Delta\theta = \frac{\lambda_{gw}}{2D}$ for the most favorable case where **k** and **p1 -p2** are perpendicular. D is the distance between the sites $D = R_{earth}\sqrt{2(1-\cos(\Omega))}$ where Ω is the angle between the z axes of the two sites. The full patch size at 100 Hz of correlation between LLO and LHO is about 28 degrees and at 1kHz it is 2.8 degrees. The corresponding values for LHO and VIRGO are 10 and 1 degrees.

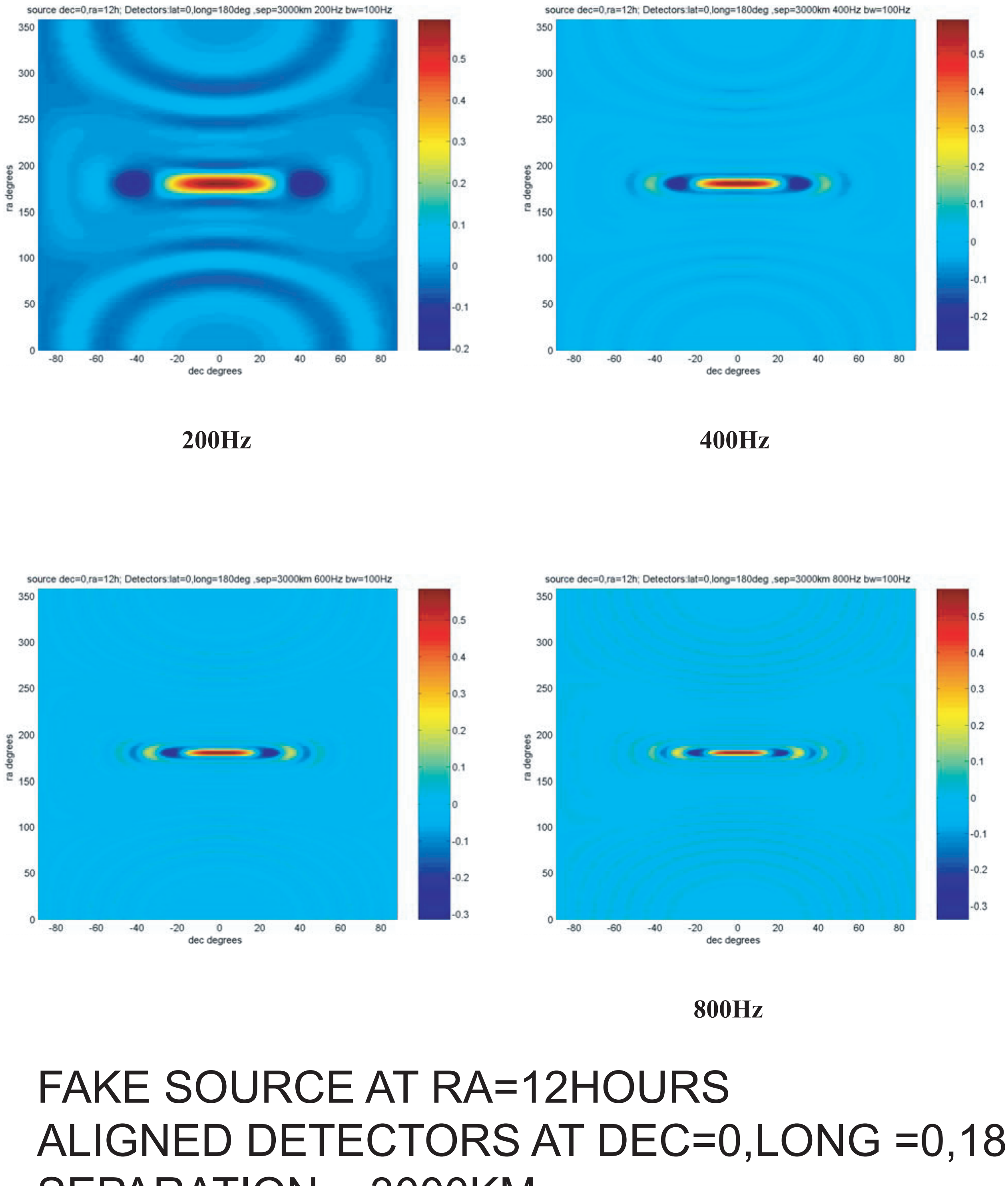
Figure 4 through **Figure 8** are "sky maps" of the beam pattern at 200,400,600 and 800 Hz for a variety of sources and detector combinations. All the maps have been made using the FORTRAN program attached in the appendix and use a detection bandwidth of 100Hz defined by a Hanning spectrum $W(f) = \frac{1}{2}(1-\cos(\frac{2\pi(f\text{-}f_0)}{\Delta B}))$ where f_0 is the center frequency and ΔB is the bandwidth. The patterns have been, furthermore, averaged over the source polarization, replacing $\sin^2(\Psi)$ and $\cos^2(\Psi)$ by 1/2 and $\sin(\Psi)\cos(\Psi)$ by 0 . In a more careful analysis, it will be one of the first tasks to examine whether this averaging is valid. It will be fine for LLO* LHO but may not be valid for LHO*VIRGO and LLO*VIRGO because of the coherent correlation of strains and the unfortuante choice of relative orientation of the arms. **Figure 4** shows the artificial case of perfect alignment. Here a source is placed at an RA of 12hours and detectors are placed with z axes in the equatorial plane at a longitude of 180 degrees with azimuth angles of 0. The two detectors are separated by 3000 km (the same as LLO and LHO) but one is on the surface of the earth while the other is buried in a remarkably hot mine.The reason for showing this case is its simplicity, the source is at one moment directly above both detectors and the maxima in the angular distribution of both detectors occur at that time. This case represents the maximum radiometer coupling efficiency to a source (around 0.6). **Figure 5** is the angular distribution pattern LLO*LHO pointing at the Galactic center (RA = 265^{o} ,dec = -29^{o}), the product of the angular distribution patterns is negative and the best radiometer coupling efficiency is approximately 0.16. **Figure 6** is the angular distribution of LHO*VIRGO pointing toward the Galactic center again with a negative product and a coupling efficiency of approximately 0.12. **Figure 9** shows higher resolution maps of the beam patterns at 800 Hz. **Figure 7** is the beam map of LLO*LHO when pointing to the Virgo cluster (RA=187^{o} , dec = 12.7^{o}), the product of angular patterns is positive and the radiometer coupling efficiency is 0.17. **Figure 8** shows the LHO*VIRGO angular distribution pattern when pointing to the Virgo cluster. The product of the angular patterns is negative and the radiometer coupling efficiency is 0.13.

Table 1: Detector coordinates

detector	latitude			longitude			azimuth
	degrees	minutes	seconds	degrees	minutes	seconds	degrees
LHO	46	27	18.53	-119	24	27.57	-35.9994
LLO	30	33	46.42	-90	46	27.26	-107.7165
VIRGO	43	37	53.09	10	30	16.19	19.4326
GEO	52	14	42.52	9	48	25.89	68.388
TAMA	35	40	35.6	139	32	9.8	-90.0(?)

Table 2: Possibly interesting sources

source	Right Ascension			Declination		distance
	hr	min	sec	deg	min	parsec
galactic center	17	42	4	-28	-55	8k
M31 Andromeda	0	40	41	41	0	670k
M81	9	51	30	69	18	3.2M
M87 Virgo cluster	12	28	18	12	40	19M



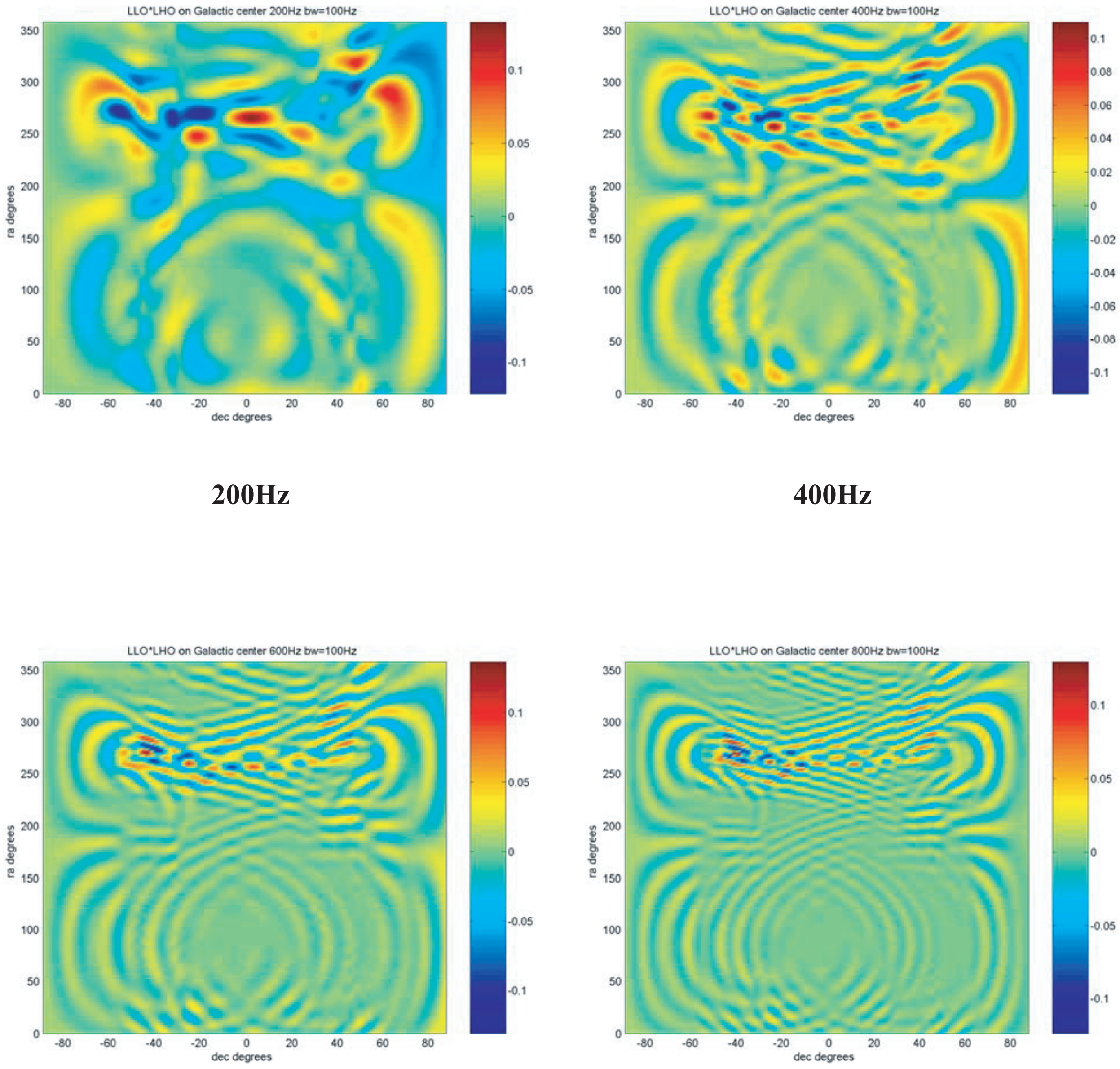
200Hz

400Hz

800Hz

FAKE SOURCE AT RA=12HOURS
ALIGNED DETECTORS AT DEC=0,LONG =0,180
SEPARATION = 3000KM

Figure 4



200Hz

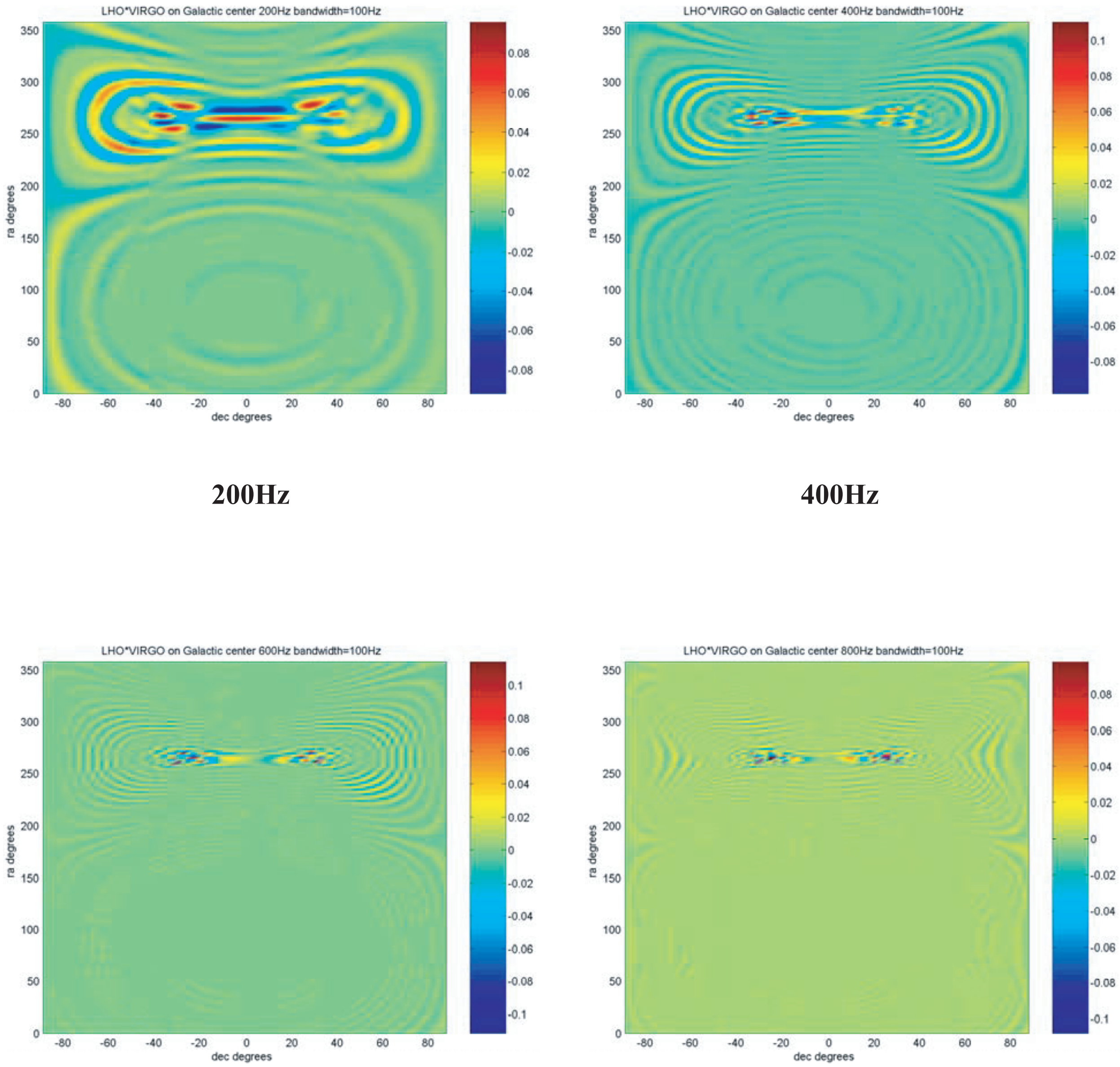
400Hz

600Hz

800Hz

LLO*LHO on the Galactic Center

Figure 5



200Hz

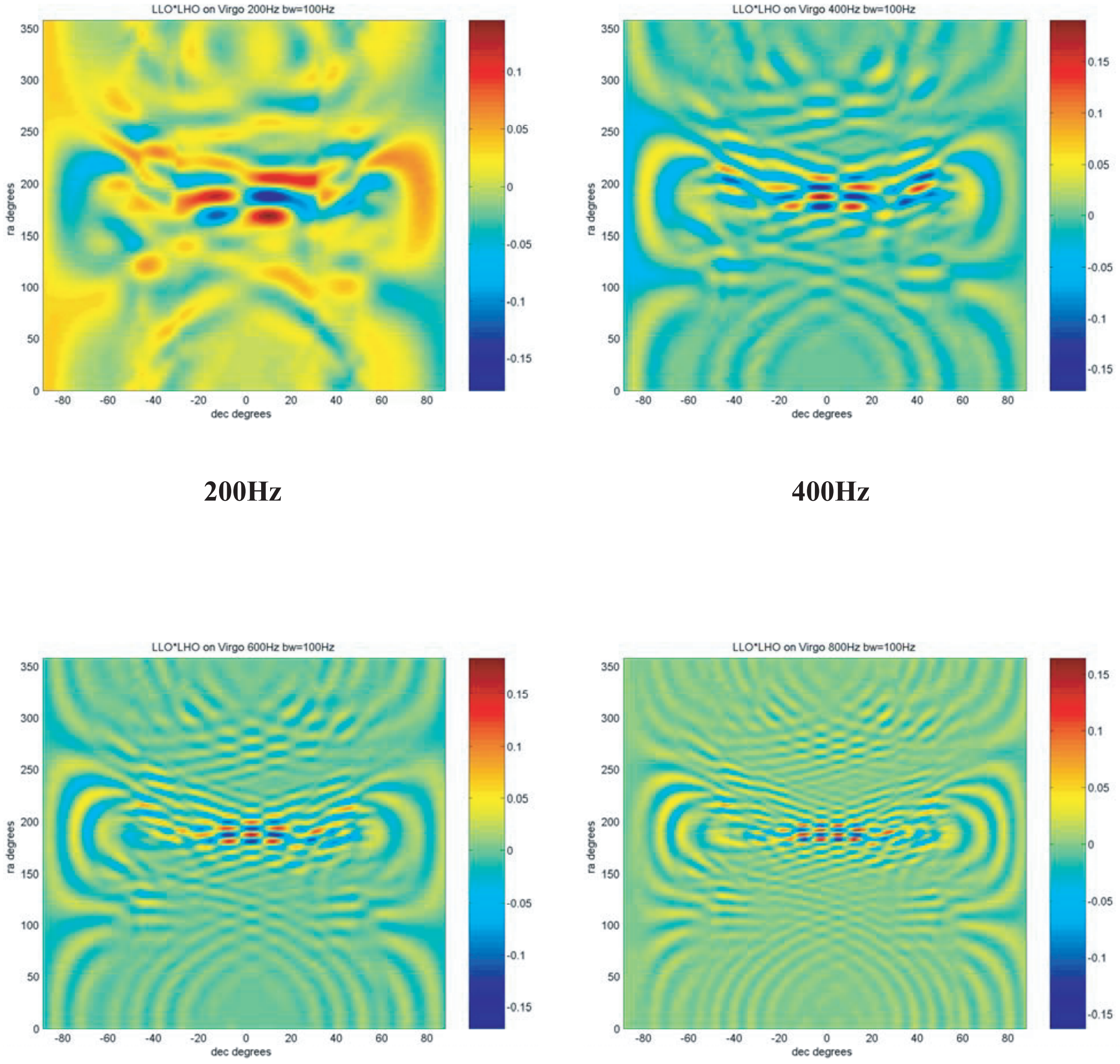
400Hz

600Hz

800Hz

LHO*VIRGO on the Galactic Center

Figure 6


LLO*LHO on Virgo 200Hz bw=100Hz

LLO*LHO on Virgo 400Hz bw=100Hz

LLO*LHO on Virgo 600Hz bw=100Hz

LLO*LHO on Virgo 800Hz bw=100Hz

200Hz

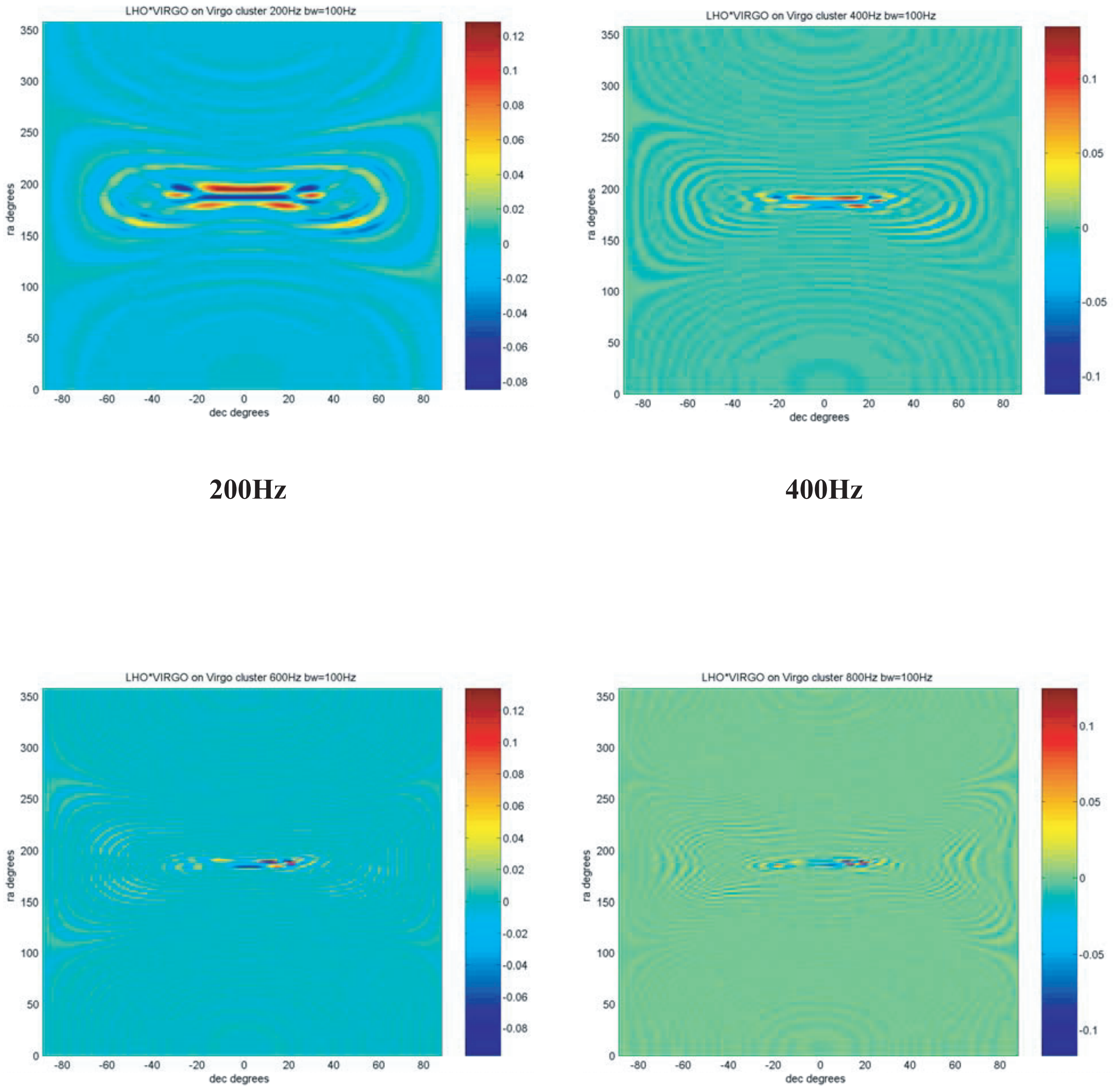
400Hz

600Hz

800Hz

LLO*LHO on Virgo clusterF

Figure 7



200Hz

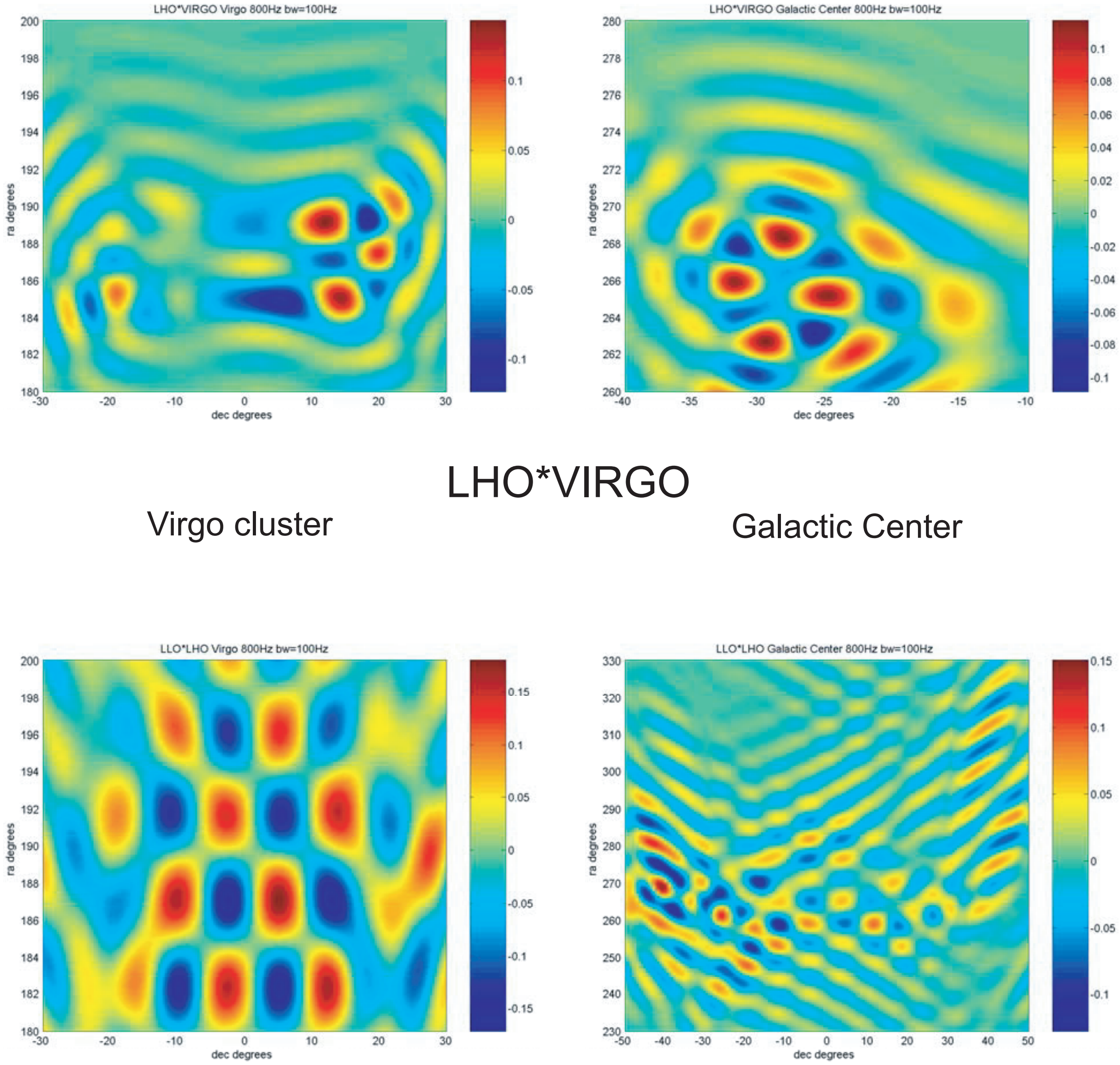
400Hz

600Hz

800Hz

LHO*VIRGO on Virgo cluster

Figure 8



LHO*VIRGO

Virgo cluster

Galactic Center

LLO*LHO

High resolution beam maps

Figure 9

Preliminary estimates of sensitivity:
A short analysis with crude assumptions is made to estimate the sensitivity of the gravitational wave radiiometer. Here again a more careful study with calculated wave forms and more thoughtful estimates of the source populations are in order.

Assume a stationary and Gaussian noise distribution in the interferometers characterized by the noise power spectrum converted to strain power density $h_n^2(f) = \sqrt{h_{n1}^2(f)h_{n2}^2(f)}$. The noise power in a single sample with bandwidth ΔB is $P_n = h_n^2(f)\Delta B = h_{rms}^2$ assuming that the spectrum is flat enough in the band. The number of independent samples of the power spectrum taken in an integration time is $N = \frac{\tau_{int}}{T_{sample}} = \tau_{int}\Delta B$. The fluctuation in the noise spectrum after averaging the N samples is $\sigma = \frac{h_{rms}^2}{\sqrt{N}} = \frac{h_{rms}^2}{\sqrt{\tau_{int}\Delta B}}$. The signal power from sources in the patch on the sky is $P_{sig} = h^2_{sigrms}\varepsilon\delta n_s$ where ε is the crosscorrelated detector power coupling efficiency, δ is the source duty cycle (for a periodic source this is unity) and n_s are the number of radiating sources that fall into the beam. Assuming that there is signal power in the band, the minimum h_{rms} that can be measured varies as

$$h_{sigrms} > h_{noiserms}\sqrt{\frac{S/N}{\varepsilon\delta n_s}}\left(\frac{1}{\tau_{int}\Delta B}\right)^{\frac{1}{4}}$$

where S/N is the desired power signal to noise. Using 100 Hz bandwidth and an integration time of 10^7 seconds, the design sensitivity at 200 Hz of $h_{noiserms} = 6 \times 10^{-22}$ and a power coupling efficiency of 0.16; the rms strain signal needs to be larger than $8 \times 10^{-24} / \sqrt{\delta n_s}$. Need to estimate the types and number of sources and their duty cycle to do more with this.

References:

Michelson, P (1987) Monthly Notices of the Royal Astronomical Society **227**, 933

Christensen, N (1990) Physics PhD Thesis MIT

Christensen, N (1992) “Measuring the stochastic gravitational-radiation background with laser-interferometric antennas” Phys Rev D **46**, 5250

Flanagan, E (1993) “Sensitivity of the Laser Interferometer Gravitational Wave Observatory to a stochastic background, and its dependence on the detector orientations” Phys Rev D **48**, 2389

Allen, B and Ottewill, A (1997) “Detection of anisotropies in the gravitational-wave stochastic background” Phys Rev D **56**, 545

Allen, B and Romano, J (1999) “ Detecting a stochastic background of gravitational radiation: Signal processing strategies and sensitivities” Phys Rev D **59**, 102001

Cornish, N (2001) “Mapping the gravitational wave background” arXiv:astro-ph/0105374v1 22 May 2001

Appendix 1 The FORTRAN program

```
c  **************** gravradiometer.for  May 19, 2004 ***************

c This version has
c 1) fixed errors in the dot product subroutine
c 2) fixed a problem in the inverse sin due to round off
c 3) added the ability to include a bandwidth with a Hann window
c 4) averaged over the polarizations properly
c
c Program will ultimately make files of the correlated response of
c LLO and LHO gravitational wave detectors to a sources in a designated
c region of the sky. The idea is to delay one of the signals by
c the travel time of the gravitational wave from the designated source
c so that the cross correlation can be made for all frequencies of
c signal from the source. The question the program is trying address
c is 1) how efficient is the radiometer to detecting the radiation from
c the source (assume random polarization of the gravitational waves)
c and 2) what is the beam pattern integrated over the rotation of the
c earth as a function of gravitational wave frequency and location on
c the sky. The intent is to chop between delays associated with the
c source position on the sky and some other point in the sky for equal
c observation time. This is a way of dealing with the systematic noise
c in the instrument as the random noise is reduced by long integration
c times.
c
c The input to the program is the RA and dec of the source
c
c The quantity being evaluated is the correlation of the two
c interferometers which is given by:
c
c   corr det1*det2 = (F1+*F2+ + F1x*F2x) * cos[k.(p1-p2) + 2*pi*f*tau]
c
c The wave from the source in inertial space is coming from the
c direction of k, the the wavevector of the wave k = (2*pi*f)/c
c where f is the frequency of the wave and c is the velocity of light.
c tau is the delay time between the data streams from
c the two detectors. (p1 - p2) is the straight line vector between the
c vertices of the two detecting sites. The F are geometric factors
c which depend on the relative orientation of the arms of the
c ground based detector and the inertial based incoming gravitational
c wave direction.
c
c  Defining the ground based gravitational wave detector x and y
c  axes along the arms and the z axis as perpendicular to the plane of
c  the detector. F factors are
```

```
c
c F+=(1/2)*(1+cos**2(theta))*(cos(2*phi)*cos(2*psi)
c     -cos(theta)*sin(2*phi)*sin(2*psi)
c
c Fx=(1/2)*(1+cos**2(theta))*(cos(2*phi)*sin(2*psi)
c     -cos(theta)*sin(2*phi)*cos(2*psi)
c
c + and x designate the polarization state of the gravitational wave,
c psi is the polarization angle around the k vector measured from the
c the phi = constant plane. theta is the
c polar angle between the z axis of the detector and
c the k vector of the incoming radiation. phi is the angle between
c the x axis of the detector and the projection of k on the x/y
c plane of the detector
use winteracter
dimension cr(100,200,10),raptdeg(200),decptdeg(100),freq(10)
dimension delay(1450)
character fileout*50
pi = acos(-1.0)
radtodeg = 180.0/pi
c the unit vectors on the earth for the two LIGO sites
xarmxlho = -0.223892
xarmylho = 0.799831
xarmzlho = 0.556905
yarmxlho = -0.913978
yarmylho = 0.0260945
yarmzlho = -0.404923
zxlho = -0.338402
zylho = -0.599658
zzlho = 0.725186
c
xarmxllo = -0.954574
xarmyllo = -0.14158
xarmzllo = -0.262189
yarmxllo = 0.297741
yarmyllo = -0.48791
yarmzllo = -0.820545
zxllo = -0.0117515
zyllo = -0.861335
zzllo = 0.507901
c
c difference vector between llo to lho
diffxllolho = -0.69397520978
diffyllolho = 0.554177464
diffzllolho = 0.45965829328
c straight line distance between sites in meters (need to check this
c number get 3110km using z1.z2 which gives 28.2546 degrees between
c detectors and earth radius = 6371 km
diffllolho =  3.030e6
c velocity of light
c = 2.9987e8
c sidereal day in seconds
siddaysec = 8.6164e4
nminsidday = int(siddaysec/60.0)
1write(unit=*,fmt=2)
2format(' MENU ' /
    &  ' 1 = choose coordinates of the source ' /
    &  ' 2 = choose range of inertial coordinates for calculation ' /
    &  ' 3 = choose gravitational wave frequency range ' /
    &  ' 4 = choose detector coordinates (default is LHO and LLO) ' /
    &  ' 5 = do the calculation ' /
    &  ' 6 = write a file of the results ' /
    &  ' 7 = end program ' /
    &  ' MODE = : ' $)
read(unit=*,fmt=3)mode
3format(i3)
go to (100,200,300,400,500,600,700)mode
100write(unit=*,fmt=101)
101format('  enter ra (hr,min,sec) of source : '$)
read(unit=*,fmt=102)sourcerahr,sourceramin,sourcerasec
102format(3e15.6)
write(unit=*,fmt=103)
103format('  enter dec (deg,min) of source : '$)
read(unit=*,fmt=104)sourcedecdeg,sourcedecmin
104format(2e15.6)
radeg = (sourcerasec/3600.0+sourceramin/60.0+sourcerahr)*360.0/24.0
decdeg = sourcedecdeg + sourcedecmin/60.0
      rarad = radeg/radtodeg
      decrad = decdeg/radtodeg
      xinertial = (cos(rarad))*(cos(decrad))
      yinertial = (sin(rarad))*(cos(decrad))
zinertial = sin(decrad)
write(unit=*,fmt=105)xinertial,yinertial,zinertial
105format('  x = '1pe15.6,' y = '1pe15.6,' z = '1pe15.6 )
go to 1
c
200write(unit=*,fmt=201)
201format('  enter # of ra pts, # of dec pts : '$)
read(unit=*,fmt=202)nra,ndec
202format(2i5)
write(unit=*,fmt=203)
203format('  enter max,min ra (deg) max, min dec (deg) : '$)
```

```
read(unit=*,fmt=204)maxradeg,minradeg,maxdecdeg,mindecdeg
204format(4e15.6)
delra = (maxradeg - minradeg)/real(nra)
deldec = (maxdecdeg - mindecdeg)/real(ndec)
do 210 j=1,nra
raptdeg(j) = minradeg + delra*real(j-1)
210continue
do 220 k=1,ndec
decptdeg(k) = mindecdeg + deldec*real(k-1)
220continue
go to 1
c
300write(unit=*,fmt=301)
301format('  enter # of frequency pts : '$)
read(unit=*,fmt=302)nfreq
302format(i5)
write(unit=*,fmt=303)
303format('  enter max, min gravitational wave frequency : '$)
read(unit=*,fmt=304)maxfreq,minfreq
304format(2e15.6)
delfreq = (maxfreq - minfreq)/real(nfreq-1)
do 310 k=1,nfreq
freq(k) = minfreq + delfreq*real(k-1)
write(unit=*,fmt=311)k,freq(k)
310continue
311format('  k = 'i4,' frequency = '1pe15.6)
write(unit=*,fmt=312)
312format('  enter averging bandwidth Hz : '$)
read(unit=*,fmt=315)deltf
315format(e15.6)
beta = 2.0*pi/deltf
go to 1
c
400write(unit=*,fmt=401)
401format('  if site 1 is llo, enter 1 : '$)
read(unit=*,fmt=402)illo
402format(i3)
if(illo.eq.1) then
xarmx1 = xarmxllo
xarmy1 = xarmyllo
xarmz1 = xarmzllo
yarmx1 = yarmxllo
yarmy1 = yarmyllo
yarmz1 = yarmzllo
zx1 = zxllo
zy1 = zyllo
```

```
zz1 = zzllo
go to 410
end if
c calculate the site unit vectors if latitude,longitude and
c the azimuth angles of the x arm is known. Assume right
c angle between x and y arms.
c longitude positive going east of Greenwich and negative
c going west of Greenwich. Latitude positive going north
c of equator, negative going south of equator
c Azimuth is measured in tangent plane with zero toward north
c and going positive toward east, negative toward west.
write(unit=*,fmt=451)
451format('  enter 1 if earth unit vectors of det1 are known : '$)
read(unit=*,fmt=452)iunit1
452format(i3)
if(iunit1.eq.1)go to 453

write(unit=*,fmt=454)
454format('  enter lat,long and azim x arm (deg) det1: '$)
read(unit=*,fmt=455)alat1,along1,aazm1
455format(3e15.6)
alat1 = alat1/radtodeg
along1 = along1/radtodeg
aazm1 = aazm1/radtodeg
call det(alat1,along1,aazm1,u11,u12,u13,v11,v12,v13,w11,w12,w13)
xarmx1=u11
xarmy1=u12
xarmz1=u13
yarmx1=v11
yarmy1=v12
yarmz1=v13
zx1=w11
zy1=w12
zz1=w13
go to 410

453write(unit=*,fmt=403)
403format('  enter xe,ye,ze of x arm of site 1 : '$)
read(unit=*,fmt=404)xarmx1,xarmy1,xarmz1
404format(3e15.6)
write(unit=*,fmt=405)
405format('  enter xe,ye,ze of y arm of site 1 : '$)
read(unit=*,fmt=404)yarmx1,yarmy1,yarmz1
write(unit=*,fmt=407)
407format('  enter xe,ye,ze of z direction of site 1 : '$)
read(unit=*,fmt=404)zx1,zy1,zz1
```

```
410write(unit=*,fmt=411)
411format('  if site 2 is lho, enter 1 : '$)
read(unit=*,fmt=402)ilho
if(ilho.eq.1)then
xarmx2 = xarmxlho
xarmy2 = xarmylho
xarmz2 = xarmzlho
yarmx2 = yarmxlho
yarmy2 = yarmylho
yarmz2 = yarmzlho
zx2 = zxlho
zy2 = zylho
zz2 = zzlho
go to 420
end if
write(unit=*,fmt=461)
461format('  enter 1 if earth unit vectors of det2 are known : '$)
read(unit=*,fmt=452)iunit2
if(iunit2.eq.1)go to 473

write(unit=*,fmt=456)
456format('  enter lat,long and azim x arm (deg) det2: '$)
read(unit=*,fmt=455)alat2,along2,aazm2
alat2 = alat2/radtodeg
along2 = along2/radtodeg
aazm2 = aazm2/radtodeg
call det(alat2,along2,aazm2,u21,u22,u23,v21,v22,v23,w21,w22,w23)
xarmx2=u21
xarmy2=u22
xarmz2=u23
yarmx2=v21
yarmy2=v22
yarmz2=v23
zx2=w21
zy2=w22
zz2=w23
       go to 491

473write(unit=*,fmt=412)
412format('  enter xe,ye,ze of x arm of site 2 : '$)
read(unit=*,fmt=404)xarmx2,xarmy2,xarmz2
write(unit=*,fmt=413)
413format('  enter xe,ye,ze of y arm of site 2 : '$)
read(unit=*,fmt=404)yarmx2,yarmy2,yarmz2
write(unit=*,fmt=414)
414format('  enter xe,ye,ze of z direction of site 2 : '$)
```

```
read(unit=*,fmt=404)zx2,zy2,zz2
420if(illo.eq.1.and.ilho.eq.1)then
diffx12=diffxllolho
diffy12=diffyllolho
diffz12=diffzllolho
dist12 = diffllolho
go to 430
end if
491write(unit=*,fmt=492)
492format(' if difference vector is known enter 1: '$)
read(unit=*,fmt=452)idiff
if(idiff.eq.1)go to 493
call vecdif(alat1,along1,alat2,along2,udx,udy,udz,sep)
diffx12=udx
diffy12=udy
diffz12=udz
dist12=sep
go to 430

493write(unit=*,fmt=421)
421format('  enter xe,ye,ze of difference 1->2 : ' $)
read(unit=*,fmt=404)diffx12,diffy12,diffz12
write(unit=*,fmt=422)
422format('  enter distance between sites in meters : '$)
read(unit=*,fmt=423)dist12
423format(e15.6)
430write(unit=*,fmt=431)xarmx1,xarmy1,xarmz1
431format(' x1 ' 1pe15.6,1pe15.6,1pe15.6)
write(unit=*,fmt=432)yarmx1,yarmy1,yarmz1
432format(' y1 ' 1pe15.6,1pe15.6,1pe15.6)
write(unit=*,fmt=433)zx1,zy1,zz1
433format(' z1 ' 1pe15.6,1pe15.6,1pe15.6)
write(unit=*,fmt=434)xarmx2,xarmy2,xarmz2
434format(' x2 ' 1pe15.6,1pe15.6,1pe15.6)
write(unit=*,fmt=435)yarmx2,yarmy2,yarmz2
435format(' y2 ' 1pe15.6,1pe15.6,1pe15.6)
write(unit=*,fmt=436)zx2,zy2,zz2
436format(' z2 ' 1pe15.6,1pe15.6,1pe15.6)
write(unit=*,fmt=437)diffx12,diffy12,diffz12,dist12
437format(' diff12 '1pe15.6,1pe15.6,1pe15.6,1pe15.6, ' meters')
go to 1
c
c THE CALCULATION
c loop over one day
c first zero the cross correlation averages
500do 502 kd=1,ndec
```

```
do 503 kr=1,nra
do 504 kf=1,nfreq
cr(kd,kr,kf) = 0.0
504continue
503continue
502continue
       mm = 0
nhour=0
amin = real(nminsidday)
do 510 kt=1,nminsidday
mm = mm + 1
if(mm.eq.60)then
nhour=nhour+1
write(unit=*,fmt=1199)nhour
1199format('  time is 'i5, ' hours')
mm = 0
end if
rotangle = 2.0*pi*60.0*(real(kt-1))/siddaysec
c rotation of site difference vector
call rotatez(diffx12,diffy12,diffz12,dxr,dyr,dzr,rotangle)
c rotation of site 1
call rotatez(xarmx1,xarmy1,xarmz1,xx1r,xy1r,xz1r,rotangle)
call rotatez(yarmx1,yarmy1,yarmz1,yx1r,yy1r,yz1r,rotangle)
call rotatez(zx1,zy1,zz1,zx1r,zy1r,zz1r,rotangle)
c rotation of site 2
call rotatez(xarmx2,xarmy2,xarmz2,xx2r,xy2r,xz2r,rotangle)
call rotatez(yarmx2,yarmy2,yarmz2,yx2r,yy2r,yz2r,rotangle)
call rotatez(zx2,zy2,zz2,zx2r,zy2r,zz2r,rotangle)
c calculate the frequency independent delay between signals from the
c two sites
      call dot(xinertial,yinertial,zinertial,dxr,dyr,dzr,coskdiff,scal)
tausource = coskdiff*dist12/c
delay(kt) = tausource
c loop over ra
do 530 kr = 1,nra
c loop over dec
do 540 kd =1, ndec
c generate unit vector to position in sky
dec = decptdeg(kd)/radtodeg
ra = raptdeg(kr)/radtodeg
xsky = cos(ra)*cos(dec)
ysky = sin(ra)*cos(dec)
zsky = sin(dec)
c establish overlap with detector sensitivity, calculate theta (angle
c between source direction and z axis of detector) and aphi (angle
c between xarm and projection of source in detector x/y plane)
```

```
c determine thetas
      call dot(xsky,ysky,zsky,zx1r,zy1r,zz1r,costheta1,sca1)
      call dot(xsky,ysky,zsky,zx2r,zy2r,zz2r,costheta2,sca2)
c determine aphis
call cross(xsky,ysky,zsky,zx1r,zy1r,zz1r,sine,akzx1,akzy1,akzz1)
      call unity(akzx1,akzy1,akzz1,akzx1,akzy1,akzz1)
      call cross(akzx1,akzy1,akzz1,xx1r,xy1r,xz1r,salx1,akzxx1,
    & akzyx1,akzzx1)
      call cross(akzx1,akzy1,akzz1,yx1r,yy1r,yz1r,saly,akzxy1,
    & akzyy1,akzzy1)
      call dot(akzxx1,akzyx1,akzzx1,zx1r,zy1r,zz1r,cx,scalar)
      call dot(akzxy1,akzyy1,akzzy1,zx1r,zy1r,zz1r,cy,scalar)
if(salx1.ge.0.99999)then
      alpha1=pi/2.0
go to 8051
end if
if(salx1.le.-0.9999)then
alpha1=-pi/2.0
go to 8051
end if
alpha1=asin(salx1)
8051if(cx.ge.0.0.and.cy.ge.0.0)then
aphi1=pi/2.0 - alpha1
end if
if(cx.le.0.0.and.cy.ge.0.0)then
aphi1=pi/2.0 + alpha1
end if
if(cx.lt.0.0.and.cy.lt.0.0)then
aphi1 = pi/2.0 + alpha1
end if
if(cx.gt.0.0.and.cy.lt.0.0)then
aphi1 = pi/2.0 - alpha1
end if
call cross(xsky,ysky,zsky,zx2r,zy2r,zz2r,sine,akzx2,akzy2,akzz2)
call unity(akzx2,akzy2,akzz2,akzx2,akzy2,akzz2)
      call cross(akzx2,akzy2,akzz2,xx2r,xy2r,xz2r,salx2,akzxx2,
    & akzyx2,akzzx2)
      call cross(akzx2,akzy2,akzz2,yx2r,yy2r,yz2r,saly,akzxy2,
    & akzyy2,akzzy2)
call dot(akzxx2,akzyx2,akzzx2,zx2r,zy2r,zz2r,cx,scalar)
call dot(akzxy2,akzyy2,akzzy2,zx2r,zy2r,zz2r,cy,scalar)
if(salx2.ge.0.9999)then
alpha2=pi/2.0
go to 8061
end if
if(salx2.le.-0.9999)then
```

```
alpha2=-pi/2.0
go to 8061
end if
alpha2=asin(salx2)
8061if(cx.ge.0.0.and.cy.ge.0.0)then
aphi2=pi/2.0 - alpha2
end if
if(cx.le.0.0.and.cy.ge.0.0)then
aphi2=pi/2.0 + alpha2
end if
if(cx.lt.0.0.and.cy.lt.0.0)then
aphi2 = pi/2.0 + alpha2
end if
if(cx.gt.0.0.and.cy.lt.0.0)then
aphi2 = pi/2.0 - alpha2
end if
c antenna patterns, average over polarizations ,psi
c assumed to be a property of the wave and the same at
c both sites
        aa1 = 0.5*(1.0+costheta1**2)*cos(2.0*aphi1)
        bb1=  costheta1*sin(2.0*aphi1)
        aa2 = 0.5*(1.0+costheta2**2)*cos(2.0*aphi2)
        bb2 = costheta2*sin(2.0*aphi2)
c calculate the crosscorrelation and sum for sky pixle
amp =  (aa1*aa2+bb1*bb2)
if(amp.gt.10.0.or.amp.lt.-10.0)then
mm = mm + 1
write(unit=*,fmt=7001)mm,amp
7001format('  cross correlation out of range 'i4,'amp= '1pe12.2)
write(unit=*,fmt=7003)kd,kr
7003format('  kd = 'i5,' kr = 'i5)
write(unit=*,fmt=7004)salx1,salx2
7004format('  salx1= '1pe12.2,' salx2= '1pe12.2)
write(unit=*,fmt=7002)alpha1,alpha2
7002format('  alpha1= '1pe12.2,' alpha2= '1pe12.2)
write(unit=*,fmt=7005)costheta1,costheta2
7005format('  costheta1='1pe12.2,' costheta2='1pe12.2)
amp = 0.0
end if
call dot(xsky,ysky,zsky,dxr,dyr,dzr,cospt,scalar)
aalpha = 2.0*pi*((dist12*cospt/c)-tausource)
c loop over frequencies and average over bandwidth
do 520 kf=1,nfreq
coef=(cos(aalpha*freq(kf)))/deltf
ac1 = (2.0*sin(aalpha*deltf/2.0))/aalpha
ac2 = (sin((aalpha-beta)*deltf/2.0))/(aalpha-beta)
```

```
ac3 = (sin((aalpha+beta)*deltf/2.0))/(aalpha+beta)
ccr = amp*coef*(ac1-ac2-ac3)
cr(kd,kr,kf) = cr(kd,kr,kf) + ccr/amin
520continue
540continue
530continue
510continue
go to 1

600write(unit=*,fmt=601)
601format(' enter output file name : ' $)
read(unit=*,fmt=602)fileout
602format(a50)
write(unit=*,fmt=603)
603format('  enter 1=corr/dec,2=corr/ra,3=delay,4=corr/ra/dec: '$)
read(unit=*,fmt=604)ipr
604format(i4)
open(unit=3,file=fileout)
if(ipr.eq.1)then
write(unit=3,fmt=605)ndec
605format(i5)
write(unit=*,fmt=606)
606format('  enter index of the gravity wave freq and ra : '$)
read(unit=*,fmt=607)kf,kr
607format(2i6)
do 610 kd = 1, ndec
write(unit=3,fmt=608)decptdeg(kd),cr(kd,kr,kf)
608format(1pe15.6,1pe15.6)
610continue
end if
if(ipr.eq.2)then
write(unit=3,fmt=605)nra
write(unit=*,fmt=611)
611format('  enter index of the gravity wave freq and dec : '$)
read(unit=*,fmt=607)kf,kd
do 620 kr = 1, nra
write(unit=3,fmt=608)raptdeg(kr),cr(kd,kr,kf)
620continue
end if
if(ipr.eq.3)then
write(unit=3,fmt= 605)nminsidday
do 630 k=1,nminsidday
xx = real(k)
write(unit=3,fmt=608)xx,delay(k)
630continue
end if
```

```
if(ipr.eq.4)then
write(unit=*,fmt=641)
641format('  enter # of frequency : '$)
read(unit=*,fmt=642)kf
642format(i4)
do 650 kd=1,ndec
do 650 kr=1,nra
write(unit=3,fmt=643)decptdeg(kd),raptdeg(kr),cr(kd,kr,kf)
643format(1pe15.6,1pe15.6,1pe15.6)
650continue
end if
close (3)
go to 1
700continue
end

subroutine dot(ax,ay,az,bx,by,bz,cosine,scalar)
scalar = ax*bx + ay*by + az*bz
suma = ax*ax + ay*ay + az*az
sumb = bx*bx + by*by + bz*bz
if(suma.eq.0.0.or.sumb.eq.0.0)then
write(unit=*,fmt=1)
1format(' dot:length of vector = 0')
end if
cosine = scalar/sqrt(suma*sumb)
return
end

subroutine cross(ax,ay,az,bx,by,bz,sine,cx,cy,cz)
suma = ax*ax + ay*ay + az*az
sumb = bx*bx + by*by + bz*bz
cx = ay*bz - az*by
cy = bx*az - ax*bz
cz = ax*by - bx*ay
sumc = cx*cx + cy*cy + cz*cz
if(suma.eq.0.0.or.sumb.eq.0.0)then
write(unit=*,fmt=1)
1format(' cross: length of vector = 0')
end if
sine = sqrt(sumc)/sqrt(suma*sumb)
return
end

subroutine unity(ax,ay,az,anx,any,anz)
sum = ax*ax + ay*ay + az*az
sum = sqrt(sum)
```

```
if(sum.eq.0.0)then
write(unit=*,fmt=1)
1format('  unity: length of vector = 0')
end if
anx = ax/sum
any = ay/sum
anz = az/sum
return
end

subroutine rotatez(ax,ay,az,arx,ary,arz,angle)
arx = ax*cos(angle) + ay*sin(angle)
ary = -ax*sin(angle) + ay*cos(angle)
arz = az
return
end

subroutine det(ala,alo,azm,u1,u2,u3,v1,v2,v3,w1,w2,w3)
c vector in tangent plane for x arm
pi = acos(-1.0)
cola = pi/2.0 - ala
xxt = cos(azm)
xyt = -sin(azm)
xzt = 0.0
c vector in tangent plane for y arm
yxt = sin(azm)
yyt = cos(azm)
        yzt = 0.0

c vector of z axis in tangent plane
zxt = 0.0
zyt = 0.0
zzt = 1.0
c define the coordinate rotation
a11 = -cos(alo)*cos(cola)
a12 = sin(alo)
a13 = cos(alo)*sin(cola)
a21 = -sin(alo)*cos(cola)
a22 = -cos(alo)
        a23 = sin(alo)*sin(cola)
a31 = sin(cola)
a32 = 0.0
a33 = cos(cola)
sx = xxt*a11 +xyt*a12 +xzt*a13
sy = xxt*a21 +xyt*a22 +xzt*a23
sz = xxt*a31 +xyt*a32 +xzt*a33
```

```
call unity(sx,sy,sz,u1,u2,u3)
sx = yxt*a11 +yyt*a12 +yzt*a13
sy = yxt*a21 +yyt*a22 +yzt*a23
sz = yxt*a31 +yyt*a32 +yzt*a33
call unity(sx,sy,sz,v1,v2,v3)
sx = zxt*a11 +zyt*a12 +zzt*a13
sy = zxt*a21 +zyt*a22 +zzt*a23
sz = zxt*a31 +zyt*a32 +zzt*a33
call unity(sx,sy,sz,w1,w2,w3)
return
end

subroutine vecdif(ala1,alo1,ala2,alo2,vx,vy,vz,sep)
re = 6.371e6
sx = cos(alo1)*cos(ala1)
sy = sin(alo1)*cos(ala1)
sz = sin(ala1)
call unity(sx,sy,sz,x1,y1,z1)
sx = cos(alo2)*cos(ala2)
sy = sin(alo2)*cos(ala2)
sz = sin(ala2)
call unity(sx,sy,sz,x2,y2,z2)
sx = x2-x1
sy = y2-y1
sz = z2-z1
call unity(sx,sy,sz,vx,vy,vz)
call dot(x1,y1,z1,x2,y2,z2,cosarc,scal)
bb = (1.0-cosarc)**2
bb = sqrt(bb)
c using law of cosines
sep = sqrt(2.0)*re*sqrt(bb)
return
end
```