Lunar Occultation Observations of B0737+270 and B0837+242

Varun Bhalerao, Shriharsh Tendulkar, Vishal Prabhu, Rohan Chabukswar

Indian Institute of Technology Bombay

under the direction of
Prof. K. Manoharan
National Center for Radio Astrophysics

December 20, 2004

Abstract

The occultations of two radio sources (B0737+270 and 0837+242) by the moon were studied using the Ooty Radio Telescope. Data from both the occultations was used to determine the exact co-ordinates of the sources. Data analysis was done using custom written codes. The procedures involved in this were studied in detail. The calculated co-ordinates were well in agreement with the known co-ordinates of the sources.

1   Data Acquisition

Data was acquired using the Ooty Radio Telescope (beam 7). The source was tracked as the moon occulted it, with a sampling interval of 20 ms. The local circumstances of the occultations were as follows.

B0737+270:		B0837+242:	
Date : 30 Nov 2004		Date : 01 Dec 2004	
Flux : 366 mJy		Flux : 698 mJy	
Immersion :		Immersion :	
Time (UT):	17:50:06	Time (UT):	23:28:17
Hour angle :	$-61.1°$	Hour angle :	$9.8°$
Position angle :	$37°$	Position angle :	$51°$
Emersion :		Emersion :	
Time (UT):	00:02:45	Time (UT):	18:42:32
Hour angle :	$-48°$	Hour angle :	$18.5°$
Position angle :	$319°$	Position angle :	$11°$

2   Data Reduction and Analysis

The raw data was converted from the binary format to a continuous ASCII format. The first processing required removal of background offset and noise from the data. Two methods were used for removing noise, each of which is described below.

The Fourier transform of the raw data was carried out (`dft.cpp` [A.1]). Due to constraint of resources, a Discrete Fourier Transform algorithm was used instead of an FFT. Next, a low pass filter was implemented using another program `noise.cpp`[A.2]. This code removed all high frequency components of the data by setting the Fourier coefficients of all frequencies greater than the specified frequency to zero. The inverse transform of this data gave the curve with all high frequency removed. Cut off frequencies of 1 Hz, 0.2 Hz and 0.05 Hz were tried. 0.05 Hz proved to be the optimal filter for the data. The second method involved taking running average of the data using bins of 250 and 500 data points. For a bin of 250 points, for a sampling interval of 20 ms, the average was over a period of 5 s. By the sampling theorem, each period requires two samples, so that the theoretical cut off frequency is 0.1 Hz. Similarly for a binning interval of 500 points, the theoretical cut off frequency is 0.05 Hz.

The noise free curves obtained from both methods agreed well. Finally, the background in both the data sets was removed by subtracting the average value of the data corresponding to the fully occulted source. The plots for all the four circumstances (immersion and emersion of both the occultations) are shown in figure.Further analysis consisted of fitting the noise-free to the fresnel diffraction curves expected from the lunar occultation of a point source. The main parameter involved in the fit is the instantaneous tangential velocity of the moon and the position angle of the occultation. Following sections describe these calculations.

3   The Lunar occultation method

3.1   Shape of the occultation curves

At meter-wavelengths and at a distance as large as distance to moon, angular size of the fresnel zones is very small compared to angular diameter of moon. Hence, moon can be considered as a straight diffracting edge and the resulting curve will be close to one produced by a point-source. The effects of irregularities in the moon's limb and the lunar ionosphere have been neglected.

While not affecting the overall curve, these irregularities will only cause fine changes in the fringe pattern.

3.2 Theory

Consider a plane wave incident on a straight edge. Also consider a co-ordinate system on the screen S with the x-axis along the straight edge and y-axis perpendicular to it. There will be no intensity variation due to diffraction along the x-axis and, therefore any arbitrary point P on the screen can be taken as $(0, y)$ without loss of generality. The origin is assumed to be on the edge of the geometrical shadow. If the x and y co-ordinates of any arbitrary point on M on the plane of the straight edge are denoted by α and β, then

$$r = \sqrt{[\alpha^2 + (\beta - y)^2 + d^2]}$$

$$\simeq d + \frac{\alpha^2 + (\beta - y)^2}{2d}$$

where d is the distance between the straight edge and the screen (topocentric distance to moon in our case). The resultant field at point P will be,

$$u(P) = \frac{-i}{\lambda} \iint \frac{Ae^{ikr}}{r} \, d\alpha \, d\beta$$

Letting $u = \sqrt{\frac{2}{\lambda d}}\alpha$, $v = \sqrt{\frac{2}{\lambda d}}(\beta - y)$ and $v_0 = \sqrt{\frac{2}{\lambda d}}y$, the integral simplifies to,

$$u(P) = \frac{1 - i}{2}u_0[\{0.5 - C(v_0)\} + i\{0.5 - S(v_0)\}]$$

where u_0 is some constant and $C(\tau)$, $S(\tau)$ are Fresnel integrals defined by,

$$C(\tau) = \int_0^\tau \cos(\frac{1}{2}\pi v^2)dv$$

$$S(\tau) = \int_0^\tau \sin(\frac{1}{2}\pi v^2)dv$$

The intensity distribution corresponding to above would be given by,

$$I(P) = \frac{1}{2}I_0[\{0.5 - C(v_0)\}^2 + \{0.5 - S(v_0)\}^2]$$

where I_0 is the intensity at the point considered, in absence of the straight edge.

3.3 Implementation

The point-source diffraction profile $(I(t))$was calculated using `fresnel.cpp`[A.3]. The time increment was taken to be 20ms,equal to the data sampling rate.

If the tangential velocity of moon be v and t be time, then we have $y = vt$. So,

$$v_0 = \sqrt{\frac{2}{\lambda d}}y = \sqrt{\frac{2\nu}{cd}}vt$$

To account for change in d during the period of occultation, we differentiate the above equation and then integrate it with respect to t. This gives,

$$v_0(t) = \sqrt{\frac{2\nu}{cd}}\left[1 - \frac{\dot{d}}{d}t\right]vt$$

2

where $\dot{d}$ is the radial velocity of moon. The value of $\dot{d}$ was estimated using the software ephemerides *Xephem* which gave the moon's topocentric distance to have changed by about 2km in 38 seconds. Hence, $\dot{d}$ was taken to be 52.63 m/s.

To account for the 4MHz bandwidth, point-source diffraction profile for 5 frequencies, centered at 327MHz, was calculated and the average profile was used for calculating positions. It was, however, observed that the radial velocity of moon and a finite telescope bandwidth did not cause a significant change in the theoretical model for a monochromatic point source, the distance to the screen being invariant.

4   Position Measurements

In the diffraction profile for a point source, putting $y = 0$ yields $I = \frac{I_0}{4}$. We use this fact to calculate the exact time of geometric occultation and hence estimate source co-ordinates to a better accuracy.

To calculate the values of I and I_0, following approach was used. The average intensity between immersion and emersion was taken as background(I_b). The average intensity long before the occultation began ((I_1) was then the intrinsic source intensity plus the background or, $I_1 = I_0 + I_b$. Rearranging,

$$I_0 = I_1 - I_b$$

The standard deviation of the data in the above regions was taken as the error in intensity values. It may be noted that I_b also includes the contribution form moon's radiation, which we assume to be constant during the occultation. Hence, the above approach gives a fair estimate for I_0.

The data was searched for all intensity values equal to $\frac{I_0}{4}$ within the calculated error bars. The corresponding time co-ordinates were noted. Using *XEphem* a software ephemerides, the topocentric co-ordinates of the center of the Moon and its angular radius were calculated at those time co-ordinates. the spatial co-ordinates were calculated. These calculations gave four possible values for the co-ordinates. The mean of these values was taken as the final result and the standard deviation of these was taken as error.

5   Results

Results of the position measurements are shown in Figure 1. The RA and declination values are quoted with the epoch 19 December 2004. These values agree well with the known values in literature.

B0737+270 :		B0837+242 :	
RA:	$7^h41^m35.6^s$	RA	$8^h39^m59.2^s$
Dec:	$27°00'56''$	Dec:	$24°03'06''$

Figure 1: Calculated source positions

5.1   Discussion on errors

The errors in the measured positions originate from the noise in the original data, irregularities in the lunar surface, refraction in the lunar ionosphere and due to the sampling time constant. It was found that the noise contributes to 0.6^s and 0.1^s of error in RA measurements of the

two sources and to $1''$ of error in the declination measurements of both the sources. The value quoted in available literature (C.Hazard, 1962) for the maximum inaccuracy due to the surface irregularities is about $2''$, corresponding to 0.12^s of RA. The inaccuracy due to the sampling are insignificant due to the small sampling time of $20ms$. It was not possible to investigate the effects of lunar ionospheric refraction. However, the effect is expected to be small and hence the errors due to it are presently neglected. In all, it is estimated that the accuracy is about 0.7^s in RA and $3''$ in declination.

6 Further Work

We aim to use this data to derive the detailed source-profile and source diameter information. Information about the angular diameter can be obtained from the slope plot at the instant of occultation.Deconvolving the observed fringe pattern with the theoretical one can yield a detailed intensity profile of the source.

7 Acknowledgements

A Appendix

This appendix contains C++ programs written to accomplish various tasks.

A.1 dft.cpp

```cpp
#include<iostream>
#include<cmath>
#include<fstream>

using namespace std;

const double t_sample = 0.02; const double; sample_num = 36400;
const double pi2 = 3.14159265359*2; const double
f_max=1/(2*t_sample); const doublef_step=f_max*2/sample_num;

int main() {
  cout.precision(3);
  cout<<"Taking DFT :"<<endl;
  double x, y, z, sums, sumc, f, m;
  ofstream fout("dft");
  for(f = -f_max;f <= f_max;f += f_step)
    {
      sums = 0;
      sumc = 0;
      cout<<"\b\b\b\b\b\b\b\b\b\b\b\"<<50+ f / 2 / f_max * 100<<"%"<<flush;
      ifstream fin("raw");
      for (int count = 0;count <sample_num;count++)
        {
        fin>>x>>y;
        m= pi2 * f * x*t_sample;
        sums += y * sin(m);
        sumc += y * cos(m);
        }
      fout<<f<<" "<<sumc<<" "<<sums<<endl;
      fin.close();
    }
  cout<<endl;
  return 0;
}
```

A.2 noise.cpp

```cpp
#include<iostream>
#include<cmath>
#include<fstream>

using namespace std;

const double t_sample = 0.02; const double sample_num = 36400;
```

```cpp
const double pi2 = 3.14159265359 * 2;

const double f_max = 1 / (2 * t_sample); const double f_step=
f_max * 2 / sample_num; const double f_noise = 0.999; const int
array_size = 1000 ; const int ar_off = array_size / 2;

int main() {
  ifstream fin("dft");
  int count = 0;
  count = 0;
  double f_real[array_size], f_imag[array_size], freq, fr, fi;
  while (fin.peek() != NULL)
  {
    fin>>freq>>fr>>fi;
    if (freq > -f_noise && freq < f_noise)
      {
       count = floor(freq / f_step + .5)+ar_off;
       f_real[count] = fr;
       f_imag[count] = fi;
      }
    if (freq > f_noise) break;
  }

  cout.precision(3);
  cout<<"Taking IDFT:"<<endl;
  double x, y, z, sums, sumc, t, m;
  ofstream fout("background_1");
  for(t = -t_sample * sample_num / 2;t <=t_sample * sample_num / 2;t +=
t_sample)    {
      sums = 0;
      sumc = 0;
      cout<<"\b\b\b\b\b\b\b\b\"<<50 + t / t_sample/ sample_num * 100<<" %
"<<flush;      for(freq = -f_noise;freq <= f_noise;freq+= f_step)
      {
      count = floor(freq/f_step+.5)+ar_off;
      m = -pi2 * t * f_step * (count - ar_off);
      sums += f_real[count] * sin(m) +
      f_imag[count] * cos(m);
      sumc += f_real[count] * cos(m) - f_imag[count] * sin(m);
      }
    fout<<t<<" "<<sumc / sample_num<<" "<<sums
        / sample_num<<endl;
  }
  cout<<endl;
  return 0;
}
```

6

A.3   fresnel.cpp

```cpp
#include<iostream>
#include<cmath>
#include<fstream>

using namespace std;

double sq(double t) {
  return t*t;
}
double fresnelc(double t) {
  double n=1000.0;
  double sum=0.0;
  double dt=t/n;
  for(double i=0;i<n;i++)
    {
      sum+=cos(M_PI*sq(t*i/n)/2.0)*(dt);
    }
  return sum;
}
double fresnels(double t) {
  double n=1000;
  double sum=0.0;
  double dt=t/n;
  for(double i=0;i<n;i++)
    {
      sum+=sin(M_PI*sq(t*i/n)/2.0)*(dt);
    }
  return sum;
}
int main() {

  int range=550;
  int T=range/2;
  float vdot=0.053758;
  float v;
  float increment=v;
  float factor=1; //1.66;
  ofstream fout("fresnel+");
  for(float t=-T;t<T;t+=0.02)
    {
      double A=fresnels(vdot*t/factor);
      double B=fresnelc(vdot*t/factor);
      fout<<(t)<<" "<<0.5*(sq(A+.5)+sq(B+.5))<<endl;
    }
  return 0;
```

```
}
```