GBT MEMO 173

The Condon Series Pointing Model in C

Don Wells*

January 6, 1998

Abstract

An implementation in the C language of Condon's Fourier Series for the "traditional" telescope pointing model is presented. The interpretation of the terms is discussed, a fit of the pointing model to the GBT structural model is presented, and the potential for determination of some of the terms of the model by metrology is reviewed.

Contents

1 Introduction 1

2 Function `csCondonSeries()` 2

3 Fitting the BFP tilt results: function `csInitTilt()` 3

4 A priori estimates of some pointing coefficients 4

5 Pointing coefficients potentially determined by metrology 6

6 Other terms which might be significant in the GBT 6

A `csCondonSeries.c` 7

B `csInclude.h` 10

Bibliography 11

1 Introduction

At first light, the GBT will be pointed using information only from its encoders and two formulæ containing about ten terms, each of which can be related to some specific imperfection of the structure. This

*mailto:dwells@nrao.edu

has often been referred to as the "traditional pointing model" in the GBT project. In GBT Memo 75 ("GBT Pointing Equations")[Con92], Condon recommended that the GBT's traditional pointing model be implemented as a two-dimensional Fourier series. He showed that the basic geometric defects of the as-built telescope (encoder zero-points, azimuth track tilt, elevation axis collimation error, telescope collimation error, gravitational bending) correspond to certain terms of the Fourier series. He also showed that the azimuth track irregularities can be represented by certain other higher-order terms of the series. In this memo, I present an implementation in C of Condon's concept. This implementation has been a part of the GBT M&C software since 1993.

The role of this implementation was defined in [HF95, p.2]:

> "We now define the dividing line between pointing effects and focus-tracking effects such that changes in the location of the prime focal point are called pointing corrections, while changes in the feedarm location will be called focus-tracking corrections. The pointing algorithm is concerned solely with controlling the Az-El drives so that the desired (refracted) target is imaged onto the point of peak gain for the primary mirror; the same pointing algorithm applies to both prime and Gregorian operation. In contrast, the two focus-tracking algorithms are concerned with manipulating the prime focus box or subreflector actuators in order to maintain the active feed at the (imaged) point of peak gain for the primary mirror."

Refraction will be computed for the GBT M&C system by the SLALIB[1] function `SLA_REFRO` (see the Web page and the code for details). This computation will depend on data from the GBT weather station(s).

The Max-Planck-Institut für Radioastronomie held a "workshop on Pointing and Pointing Models" on March 7 and 8, 1996, and the proceedings were published as Technischer Bericht Nr. 78 of MPIfR. There were 15 presentations, covering a wide variety of topics, from generalities to observed behavior of specific telescopes (Effelsberg 100-m, IRAM 30-m, Plateau de Bure 15-m, HHT, JCMT). Several of these presentations [Alt96, Nei96] are referenced below.

2 Function csCondonSeries()

The interface to this function is

```
void csCondonSeries (                   /* no return value              */
                     double az,         /* Azimuth   input        (rad) */
                     double el,         /* Elevation input        (rad) */
                     double *daz,       /* Az correction result   (rad) */
                     double *del,       /* El correction result   (rad) */
                     double *daz_track, /* Az-track component     (rad) */
                     double *del_track, /* Az-track component     (rad) */
                     struct CsCondonCoeff *az_series,
                     struct CsCondonCoeff *el_series)
```

The input variables of the function are `az` and `el`, and the coefficients enter via `az_series` and `el_series`. The pointing correction returns in variables `daz` and `del`; note that the value returned in `daz` is actually $\Delta A \cos E$, so it must be divided by $\cos E$. Variables `daz_track` and `del_track` were provided in the design of this function circa 1993 to facilitate prediction of the alidade autocollimator outputs; subsequently autocollimator development was terminated.

The code for this function is shown in Appendix A (p.7) and the definition of `struct CsCondonCoeff` is shown in Appendix B (p.10). Terms up to – and beyond – `MAXORDER=2` are included in the code,

[1] See `http://info.gb.nrao.edu/GBT/MC/src/libraries/Starlink/doc/MainIndex.html`.

$\Delta A \cos E$ (Azimuth pointing function)

csCondonSeries()								
coeff	term	meaning	[Con92]	[vH93]	[Stu72]	[Mur92]	[Cla74][a]	[Nei96][b]
$d_{0,0}$	1	horizontal coll.	C_1	P_5	P_1	C_9	A_6	P_1[c]
$b_{0,1}$	$\sin E$	elev. axle coll.	C_3	P_3	P_3	C_8	A_5	P_3
$d_{0,1}$	$\cos E$	azimuth zero	C_2	P_6	P_2	C_5	A_7	P_2
$a_{1,1}$	$\cos A \sin E$	zenith E tilt	C_4	P_2	P_4	C_6	A_1	P_4
$b_{1,1}$	$\sin A \sin E$	zenith N tilt	C_5	P_1	P_5	C_7	A_2	P_5
$c_{2,1}$	$\sin 2A \cos E$	azimuth track						
$d_{2,1}$	$\cos 2A \cos E$	azimuth track						

ΔE (Elevation pointing function)

csCondonSeries()								
coeff	term	meaning	[Con92]	[vH93]	[Stu72]	[Mur92]	[Cla74]	[Nei96]
$d_{0,0}$	1	elevation zero	C_6	P_4	P_7	C_1	D_5	P_7
$c_{1,0}$	$\sin A$	zenith E tilt	C_4	P_2	P_4	C_3	D_1	P_4
$d_{1,0}$	$\cos A$	zenith N tilt	C_5	P_1	P_5	C_2	D_2	P_5
$b_{0,1}$	$\sin E$	asym. gravity		P_8			D_4	"G"
$d_{0,1}$	$\cos E$	sym. gravity	C_7	P_7	P_8	C_4	D_3	P_8

[a] The VLA formula for $\Delta A \cos E$ includes the terms $A_3 \cos A \cos E$ and $A_4 \sin A \cos E$, which correspond to the inactive GBT terms `D(az_series,1,1)` and `C(az_series,1,1)`. These terms represent the azimuth encoder "centering error", with period one turn. The analogous terms in elevation have the same form as the two gravity bending terms.

Some of the VLA antennas are known to exhibit significant 3*A* terms in their pointing (VLA antennas have *three* azimuthal support points), but these terms have not been implemented in the VLA pointing formula because high frequency operations are now done by frequent reference source offsets, a procedure which automatically corrects for these terms [Ken Sowinsky, private communication].

[b] Neidhöfer [Nei96] includes Stumpff's P_6 terms in the current Effelsberg 100-m implementation of $\Delta A \cos E$ and ΔE, with amplitude of order ±10 microradians. (See discussion in Sect. 6.)

[c] Neidhöfer [Nei96] calls this term "P_{10}", and gives a formula for it which includes P_1 plus terms for daily empirical zero point corrections, receiver offset and polarization effects. He gives analogous formulæ for P_2 and P_7. He states that these are the formulæ which are implemented on the MPIfR 100-m.

Table 1: Comparison of pointing formulæ notations

but the ones which are expected to be zero (because they are non-physical) are commented-out by the "`#ifdef EXTRA`" tests. The terms are coded as macro invocations, which expand to expressions which depend on on the sines and cosines of the azimuth and elevation. The final result of this expansion is displayed in Figure 3 (p.8). We see that twelve Condon Series coefficients are active in the implementation. However, only ten are truly independent: the terms labelled as "zenith E tilt" and "zenith N tilt" in Table 1 are shared between the `az_series` and `el_series` (i.e., `C(el_series,1,0)`≡`A(az_series,1,1)` and `D(el_series,1,0)`≡`B(az_series,1,1)`). The relationship between these ten Condon terms and the terms in several prior "traditional models" is shown in Table 1.

The Condon Series is *not* orthogonal. More precisely, it is only partially orthogonal, because the telescope cannot observe below the horizon. Condon discusses this situation in the second paragraph on p.9 of [Con92], and tabulates the correlation coefficients in his Table 1 on p.14.

3 Fitting the BFP tilt results: function `csInitTilt()`

The GBT structural model [WK95b], via the best-fitting-paraboloid analysis [WK95a], is able to provide predicted values for the gravitational bending terms `B(el_series,0,1)` and `D(el_series,0,1)` in the

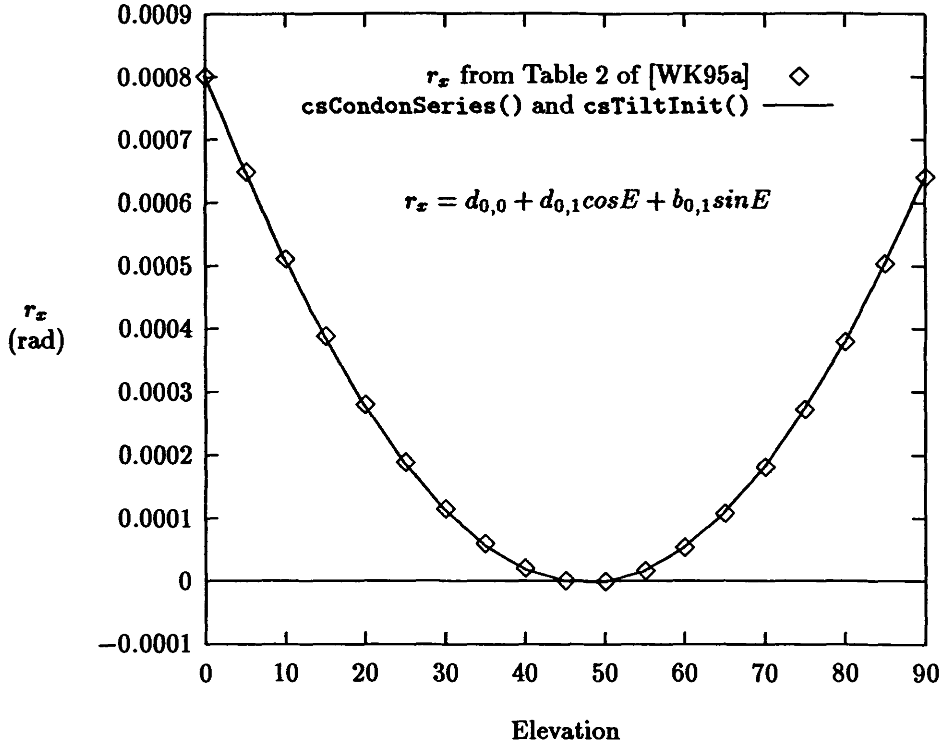

Figure 1: Sine/cosine fitted to gravitational tilt from [WK95a]

Condon series for ΔE. The predicted tilt from the BFP has been fitted with these Condon series terms, and the results used to generate function `csInitTilt()` (see Figure 2). This function will be invoked during initialization of the M&C system. The pointing/focus-tracking calibration procedures will only need to solve for differential correction terms, especially zero point shifts.

Figure 1 shows the data points from the BFP analysis and the ΔE curve computed by `csCondonSeries()` using coefficients initialized by the generated function `csInitTilt()`. The standard errors of the fitted coefficients and of the overall fit are shown as comment text in Figure 2; the standard error of fit of the function to the data values is about one microradian (0.2 arcseconds). The correlation matrix of the coefficients is also shown; the correlations are nearly unity (i.e., the three coefficients are linearly dependent; the correlation between $b_{0,1}$ and $d_{0,1}$ is 0.64 [Con92, Table 1]).

4 A priori estimates of some pointing coefficients

Waterlevel measurements made after construction of the azimuth track show that the zenith vector of the GBT is likely to be collinear with the gravity vector to about 10 microradians (2 arcseconds). This determination bounds the sum of the azimuth track irregularity terms `C(az_series,2,1)` and `D(az_series,2,1)`, plus any others which we may activate to fit observational data. (Neidhöfer [Nei96, p.31] quotes about 30 microradians shift between the instrumental zenith and the real one for the MPIfR 100-m.)

Measurements which have been made on the GBT elevation bearings are also encouraging; they suggest that the elevation axle collimation term may be as small as 20 microradians. (Neidhöfer [Nei96, p.31] states that the elevation axis collimation error for the secondary focus of the MPIfR 100-m is about 10 microradians.)

```
/* csInitTilt.c -- initialize gravitational terms of Condon Series

   Computed by csFitTilt.awk at 1997-12-31T20:13:47Z from
   BFP analysis of "MOD95.DAT GRAVITY LOADS (ZEN/HOR)
                    OCTOBER 28, 1993 MSC/NASTRAN 4/15/93" */

[GNU GPL copyright notice omitted]

#include "csInclude.h"

void csInitTilt(          /* no result value */
                struct CsCondonCoeff *az_series,
                struct CsCondonCoeff *el_series)
{
  int i, j;

  for (i=0; i<=MAXORDER; i++) { for (j=0; j<=MAXORDER; j++) {
    az_series->a[i][j] = az_series->b[i][j] =
    az_series->c[i][j] = az_series->d[i][j] =
    el_series->a[i][j] = el_series->b[i][j] =
    el_series->c[i][j] = el_series->d[i][j] = 0.0;
  } }

  el_series->d[0][0] =  0.0024615; /* elevation zero point (rad) */
                 /* +/- 0.0000024 */

  el_series->d[0][1] = -0.0016600; /*  symmetric gravity   (rad) */
                 /* +/- 0.0000020 */

  el_series->b[0][1] = -0.0018223; /* asymmetric gravity   (rad) */
                 /* +/- 0.0000020 */

     /* sigma-of-fit =  0.0000012 */

/*  Correlation matrix:
              d00   d01   b01
           +----- ----- -----
       d00 | 1.00 -0.97 -0.97
       d01 |       1.00  0.91
       b01 |             1.00   */

  return;
}
```

Figure 2: Fitted gravitation bending: function `csInitTilt()`

Note that temperature differences of only a few degrees between the alidade towers will produce variations in the axle collimation error of comparable amplitude.

5 Pointing coefficients potentially determined by metrology

Several of the pointing coefficients of the GBT are likely to be determined before first light:

- The azimuth track irregularity terms, `C(az_series,2,1)` and `D(az_series,2,1)`, can be calibrated by making laser rangefinder measurements on the alidade as a function of azimuth. High quality data of this type is likely to also allow determination of higher-ordered terms with `MAXORDER=3` or more, as discussed in Section 6 (also see the discussion in [Gol97, pp.69–70]).
- The elevation axle collimation error C_3, `B(az_series,0,1)`, can be determined by making laser rangefinder measurements to retroreflectors on the tipping structure as a function of both azimuth and elevation (see the discussion in [Gol97, pp.69–70]).
- If the true azimuth of the line connecting any two of the ground-based rangefinders is known, the azimuth zero point term C_2 (`D(az_series,0,1)`) can be determined by observing a retroreflector as a function of elevation.
- The metrology system will provide a good check, and perhaps even an improvement, of the structural model's predictions of the gravitational bending coefficients (Section 3).

These determinations of coefficients by metrology are likely to be more accurate than any that we could get from observing radio sources. This will permit holding these terms fixed while the remaining terms are determined from observation.

6 Other terms which might be significant in the GBT

The GBT is likely to have significant elevation terms due to the azimuth track. Such terms were not discussed in [Con92], because the symmetry of the geometry used in the derivation implied very little net twisting of the elevation axle due to the azimuth track. However, the GBT has only one elevation encoder, and it is attached to one side of the alidade. Azimuth track variations will twist that side of the alidade, producing changes of the encoder readings and the servo will change the pointing of the telescope in response to these changes.

It is conceivable, although unlikely, that the GBT might display a small pointing change in azimuth as a function of elevation, due to the "elevator asymmetry" of the feedarm.

There must be a $4A$ variation of the vertical stiffness of the GBT foundation, due to the four radial ribs in the foundataion, but this should cause very little variation of pointing in the GBT because the GBT is supported by four trucks: all four trucks will move vertically with the variation of stiffness, by about the same amount, and so the alidade should not tilt. If some residual angular variation does appear, we can increase `MAXORDER` to 4 and uncomment terms as needed.

Several authors [Wen90, Alt96] have asserted that the Bonn 100m needs higher-order terms to model its gravitational bending. For example, [Wen90] includes the term $C_7 \cos^3 E$ in the ΔE formula and calls this the "2^{nd} order elevation term".[2] However, this may have been due to an incorrect estimate of the beam deviation factor (BDF), as discussed by Altenhoff [Alt96, p.29]. The current 100m implementation includes

[2] At first glance the term $\cos^3 E$ does not seem to occur in the Condon Series, but in fact it is present: J.Condon [private communication] points out that $\cos^3 E = \frac{1}{4}\cos 3E + \frac{3}{4}\cos E$, so $d_{0,3}$ plus $d_{0,1}$ represents it. Condon also notes that any potential (small) nonlinear deviation from Hooke's Law would appear first as $\cos 2E$, not as $\cos 3E$.

the asymmetric gravitational term [Nei96], and presumably it is now representing the deviations from pure $\cos E$ dependency.

Neidhöfer [Nei96] indicates that the $P_6 \sin A$ and $P_6 \sin E \cos A$ terms are nonzero in the current MPIfR 100-m model; the corresponding terms in the GBT implementation would be `C(az_series,1,0)` and `B(el_series,1,1)`, which are currently disabled by "`#if EXTRA`" (see Appendix A). Stumpff [Stu72, p.435] introduced these terms, which represent an offset of the declination of a *single* calibrator source. Coefficient $P_6 = -d\delta \cos\phi / \cos\delta$ [Stu72, Eq.11,p.435] is *not* a constant–it represents the error $d\delta$ of the declination δ of a particular source, and therefore is unsuitable for fitting to observations of a *set* of sources. Stumpff says [Stu72, p.436] "..since the declination error enters the measured position difference independent of all other instrument errors, one can determine it by astronomical observations and thus measure the absolute declination of the radio source.. this formal possibility.. has no astronomical meaning because source positions today are generally known already more exactly than necessary..".

A csCondonSeries.c

[NOTE: Figure 3 (p.8) shows this code after the C preprocessor has expanded all of the macro references.]

```
/* csCondonSeries.c --- function to evaluate the Condon Series
   D.Wells, NRAO-CV
   1993-??-??: original version
   1997-05-03: many name changes, csInclude.h created
   1997-06-18: removed 'l' from 'g' format
   1997-12-16: az_series & el_series now passed as formal parameters
   1997-12-22: delete TEMPVARS and NTHETA macros, revise comments
   1997-12-30: add $4 A$ macros, C(az_series,3,0), D(az_series,3,0)
   1998-01-02: many minor revisions to comments and code
   */
```

[GNU GPL copyright notice omitted]

```
#include <math.h>
#include "csInclude.h"
```

The Condon-Series terms are defined in Table-1 and Equation 7 of GBT Memo 75 "GBT Pointing Equations" [Con92]. The classic 7-coefficient pointing formula can be represented by the terms up to $b_{1,1}$ in azimuth and $d_{0,1}$ in elevation. Only 10 of the 14 series terms are used, and two of those, C_4 (`A(az_series,1,1)`) and C_5 (`B(az_series,1,1)`), are used twice, once for azimuth and once for elevation. In [Con92] it is shown that terms can be added to represent the azimuth track irregularities. This implementation generates the sum of a list of terms by expanding a set of C macros. The sine and cosine of the azimuth and of the elevation are computed once and then are used in all of the series terms. Additional temporary variables hold integer powers of the sine and cosine. Trigonometric identities `S1` through `C4` are used to expand the Fourier terms in $n\theta$ into expressions which are functions of θ. Four macros compute the $a_{i,j}$, $b_{i,j}$, $c_{i,j}$ and $d_{i,j}$ terms of Eq.(7) of [Con92], which are products of sines and cosines in $n\theta$. The intent of this set of macro transformations is to minimize redundant operations and to give the compiler maximum opportunity for recognition of common subexpressions. The code below activates only the 10 "physical" terms. If macro symbol `EXTRA` is defined, it will activate the remaining (generally zero-coefficient) terms. `MAXORDER=2` (defined in `csInclude.h`) is sufficient for the azimuth track representation Condon chose for the Bonn 100m (see equation-14 and associated discussion in [Con92, p.11]). See Equation 14 of [Con92] for additional azimuth terms associated with the azimuth track irregularities; the leading omitted terms are `C(az_series,3,0)` and `D(az_series,3,0)` (these will need `MAXORDER=3`). Additional temporary variables analogous to `sin2_az` should be added as `MAXORDER` is increased.

```
void csCondonSeries(
                    double az,
                    double el,
                    double *daz,
                    double *del,
                    double *daz_track,
                    double *del_track,
                    struct CsCondonCoeff *az_series,
                    struct CsCondonCoeff *el_series)
{
  double sin_az, cos_az, sin_el, cos_el, sin2_az, cos2_az;
  sin_az = sin(az);
  cos_az = cos(az);
  sin2_az = sin_az*sin_az;
  cos2_az = cos_az*cos_az;
  sin_el = sin(el);
  cos_el = cos(el);
  *daz =
      az_series->d[0][0]
    + az_series->b[0][1]*sin_el
    + az_series->d[0][1]*cos_el
    + az_series->a[1][1]*sin_az*sin_el
    + az_series->b[1][1]*cos_az*sin_el
    ;
  *daz +=(*daz_track =
    + az_series->c[2][1]*2.0*sin_az*cos_az*cos_el
    + az_series->d[2][1]*(cos2_az - sin2_az)*cos_el
    );
  *del =
      el_series->d[0][0]
    + el_series->c[1][0]*sin_az
    + el_series->d[1][0]*cos_az
    + el_series->b[0][1]*sin_el
    + el_series->d[0][1]*cos_el
    ;
  *del +=(*del_track = 0.0
    );
  return;
}
```

Figure 3: Condon Series source code *as compiled*

```
#define SIN(t)  sin_##t
#define COS(t)  cos_##t
#define SIN2(t) sin2_##t
#define COS2(t) cos2_##t
#define SIN3(t) sin3_##t
#define COS3(t) cos3_##t
#define SIN4(t) sin4_##t
#define COS4(t) cos4_##t

#define S1(t) SIN(t)
#define S2(t) 2.0*SIN(t)*COS(t)
#define S3(t) (3.0*SIN(t) - 4.0*SIN3(t))
#define S4(t) (4.0*SIN(t)*COS(T) - 8.0*SIN3(t)*COS(t))
#define C0(t) 1.0
#define C1(t) COS(t)
#define C2(t) (COS2(t) - SIN2(t))
#define C3(t) (4.0*COS3(t) - 3.0*COS(t))
#define C4(t) (8.0*COS4(t) - 8.0*COS2(t) + 1.0)

#define A(v,p,q) v##->##a[p][q] * S##p(az) * S##q(el)
#define B(v,p,q) v##->##b[p][q] * C##p(az) * S##q(el)
#define C(v,p,q) v##->##c[p][q] * S##p(az) * C##q(el)
#define D(v,p,q) v##->##d[p][q] * C##p(az) * C##q(el)

#undef EXTRA

void csCondonSeries (                    /* no return value            */
                     double az,          /* Azimuth   input      (rad) */
                     double el,          /* Elevation input      (rad) */
                     double *daz,        /* Az correction result (rad) */
                     double *del,        /* El correction result (rad) */
                     double *daz_track,  /* Az-track component   (rad) */
                     double *del_track,  /* Az-track component   (rad) */
                     struct CsCondonCoeff *az_series,
                     struct CsCondonCoeff *el_series)
{
  double sin_az, cos_az, sin_el, cos_el, sin2_az, cos2_az;

  sin_az = sin (az);
  cos_az = cos (az);
  sin2_az = sin_az * sin_az;
  cos2_az = cos_az * cos_az;
  sin_el = sin (el);
  cos_el = cos (el);

  *daz =                /* Azimuth series: */
      D(az_series,0,0)  /* C_1 horizontal telescope collimation error */
#if EXTRA
    + A(az_series,0,1)
    + C(az_series,1,0) /* (source declination error) */
    + D(az_series,1,0)
#endif
    + B(az_series,0,1) /* C_3 tipping mount collimation error */
    + D(az_series,0,1) /* C_2 azimuth zero point */
    + A(az_series,1,1) /* C_4 azimuth track tilt */
    + B(az_series,1,1) /* C_5 azimuth track tilt */
    ;
  *daz += (*daz_track =
#if EXTRA
    + C(az_series,1,1) /* azimuth encoder centering error */
    + D(az_series,1,1) /* azimuth encoder centering error */
    + C(az_series,2,0)
    + D(az_series,2,0)
    + B(az_series,0,2)
    + D(az_series,0,2)
    + A(az_series,2,1)
    + B(az_series,2,1)
#endif
    + C(az_series,2,1)  /* Azimuth track irregularity */
    + D(az_series,2,1)  /* Azimuth track irregularity */
#if EXTRA
    + C(az_series,3,0)  /* Azimuth track irregularity */
    + D(az_series,3,0)  /* Azimuth track irregularity */
#endif
    ) ;

  *del =               /* Elevation series: */
      D(el_series,0,0) /* C_6 vertical telescope collimation error   */
    + C(el_series,1,0) /* C_4 azimuth track tilt (same as for 'daz') */
    + D(el_series,1,0) /* C_5 azimuth track tilt (same as for 'daz') */
    + B(el_series,0,1) /* GBT needs assymmetric gravitational bending! */
    + D(el_series,0,1) /* C_7 symmetric term of gravitational bending */
#if EXTRA
    + A(el_series,1,1)
    + B(el_series,1,1) /* (source declination error) */
#endif
    ;

  *del += (*del_track = 0.0
#if EXTRA
    + C(el_series,1,1)
    + D(el_series,1,1)
    + C(el_series,2,0)
    + D(el_series,2,0)
    + B(el_series,0,2)
    + D(el_series,0,2)
    + A(el_series,2,1)
    + B(el_series,2,1)
    + C(el_series,2,1)
    + D(el_series,2,1)
#endif
    ) ;

  return;
}
```

B csInclude.h

```
/* csInclude.h --- include file for the Condon Series code
   D.Wells, NRAO-CV
   1993-??-??: original version
   1997-05-03: many name changes
```

```
  1997-06-24: added csCondonCoeff struct type & #ifdef CSCONDON_MAIN for globals
  1997-12-16: az_series & el_series now passed as formal parameters
  */
```

[GNU GPL copyright notice omitted]

```
#ifndef CS_INCLUDE_H
#define CS_INCLUDE_H

#define MAXORDER 2
struct CsCondonCoeff {
  double a[MAXORDER+1][MAXORDER+1];   /* sin{p A} sin{q E} */
  double b[MAXORDER+1][MAXORDER+1];   /* cos{p A} sin{q E} */
  double c[MAXORDER+1][MAXORDER+1];   /* sin{p A} cos{q E} */
  double d[MAXORDER+1][MAXORDER+1];   /* cos{p A} cos{q E} */
};

/* -=-=-=-=-=-=-=- Function Prototypes: -=-=-=-=-=-=-=- */
#ifdef __cplusplus
extern "C" {
#endif /* endif cplusplus */

void csCondonSeries (                   /* no return value            */
                     double az,         /* Azimuth   input      (rad) */
                     double el,         /* Elevation input      (rad) */
                     double *daz,       /* Az correction result (rad) */
                     double *del,       /* El correction result (rad) */
                     double *daz_track, /* Az-track component   (rad) */
                     double *del_track, /* Az-track component   (rad) */
                     struct CsCondonCoeff *az_series,
                     struct CsCondonCoeff *el_series)
;

void csInitTilt(
                struct CsCondonCoeff *ax_series,
                struct CsCondonCoeff *el_series)
;

#ifdef __cplusplus
};
#endif /* endif cplusplus */
#endif /* CS_INCLUDE_H */
```

References

[Alt96] Wilhelm J. Altenhoff. Search for systematic errors and limitations. In W. J. Altenhoff, editor, *Workshop on Pointing and Pointing Models*, number 78 in MPIfR Technischer Bericht, pages 27–30. Max-Planck-Institute für Radioastronomie, 1996. This paper discusses the history of pointing investigations at the Effelsberg 100-m, relating them to various changes made in the telescope hardware. A number of incompletely understood phenomena and anomalous events are described.

[Cla74] B. G. Clark. Geometry routines – GEOM10, GEOMLR, GEOMA, GEOMDL. VLA Computer Memo 112, National Radio Astronomy Observatory, July 1974. The pointing formula used with VLA antennas is described on page 5 of this memo.

[Con92] J. J. Condon. GBT pointing equations. GBT Memo 75, National Radio Astronomy Observatory, 1992. Recommends that the GBT pointing coefficients be implemented as a 2D Fourier series.

[Gol97] M. A. Goldman. GBT coordinates and coordinate transformations. GBT Memo 165, National Radio Astronomy Observatory, February 1997.

[HF95] D. E. Hogg and J. R. Fisher. A revised description of the GBT pointing system. GBT Memo 122, National Radio Astronomy Observatory, February 1995. Replaces GBT Memo 103. Defines the "focus tracking" concept.

[Mur92] Thomas Murphy. Pointing accuracy of the 45-foot antenna. OVLBI-ES Memo 31, National Radio Astronomy Observatory, July 1992.

[Nei96] J. Neidhöfer. The actual pointing of the 100-m telescope. In W. J. Altenhoff, editor, *Workshop on Pointing and Pointing Models*, number 78 in MPIfR Technischer Bericht, pages 31–34. Max-Planck-Institute für Radioastronomie, 1996. This paper describes the pointing implementation of the MPIfR 100-m telescope as it was in March 1996. It gives current values for the coefficients, and discusses operational procedures and various anomalies that are seen.

[Stu72] Peter Stumpff. Astronomische pointingtheorie für radioteleskope. *Kleinheubacher Berichte*, 15:431–437, 1972. ("Astronomical pointing theory for radio telescopes" was translated to English by Carol Hollis and Peter Stumpff, November 1978.).

[vH93] Sebastian von Hoerner. Astronomical pointing parameters. GBT Memo 110, National Radio Astronomy Observatory, November 1993. The traditional pointing model is discussed, with emphasis on the physical significance of the terms, measurement procedures and mean errors of the coefficients.

[Wen90] Wu Wenda. The pointing correction of the 100m radio telescope in Max-Planck-Institute für Radioastronomie. *Publ. Beijing Astronomical Observatory*, 15:65–68, 1990.

[WK95a] Don Wells and Lee King. GBT Best-Fitting Paraboloid [BFP] in C. GBT Memo 131, NRAO, June 1995. Abstract: The gravitational displacements of the GBT actuators have been fitted with a paraboloid. The parameters of the paraboloid for various elevations have been fitted with polynomials and expressed as C code which computes the parameters of this best-fitting-paraboloid [BFP] as a function of elevation. The BFP will be used by the control software modules for the pointing, focus-tracking and active-surface subsystems of the GBT. We give a description of this C-code version of the BFP and two examples of its application to practical problems. We also give a function in C which fetches node data from the structural model and transforms it to a coordinate system tied to the BFP. The predicted gravitational term of the GBT's traditional pointing model and the predicted prime focus focus-tracking formula of the GBT are given. See `ftp://fits.cv.nrao.edu/pub/gbt_dwells_doc.tar.gz` for the current revision of this memo (131.2 as of 1997-06-23).

[WK95b] Don Wells and Lee King. The GBT Tipping-Structure Model in C. GBT Memo 124, NRAO, March 1995. Abstract: The finite element model of the GBT tipping structure has been translated into executable code expressed in the C language, so that it can be used by the control software modules for the pointing, focus-tracking, quadrant detector, active-surface and laser-rangefinder subsystems of the GBT. We give a description of this C-code version of the tipping structure model and two examples of its application to practical problems. See `ftp://fits.cv.nrao.edu/pub/gbt_dwells_doc.tar.gz` for the current revision of this memo (124.3 as of 1997-06-23).