Polarization Calibration of Linear Feeds – Keeping Linear Feed Basis

W. D. Cotton (NRAO), DRAFT September 6, 2023

***Abstract*—Obit has traditionally transformed data from linearly polarized feeds to a circular basis when applying polarization calibration. This memo describes the modification to keep data in the linear basis.**

***Index Terms*—Polarization calibration, Linear feeds.**

I. Introduction

TRADITIONALLY Obit was unable to image Q and U data from data with a linear feed basis (XX,YY,XY,YX); as a result, the application of polarization calibration transformed the data to a circular basis (RR,LL,RL,LR). The imaging restriction has been relaxed and the polarization calibration now has an option to leave the data in the linear basis after applying calibration. Polarization calibration of circular and linear basis data using the Obit [1][1] package is described in [2], [3]. In the nomenclature used here, the horizontal (H) feeds are referred to as "X" and the vertical (V) feeds as "Y".

II. Transformation From Circular to Linear Basis

Polarization calibration of linear feed data is discussed in [3]. The Muller matrix used to correct the on–axis instrumental polarization response to data in a circular basis is given in the Appendix of that memo. The modification allowing the output of polarization calibration to remain in a linear basis is to replace the correction of the RL and LR correlations for the parallactic angle with a transformation to the linear basis.

III. Implementation in Obit

The circular to linear transformation uses the ObitUVCalPolarization class member Cir2Lin which is given in the Appendix along with the routine to calculate the inverse Jones matrix for perfect circular feeds. In the user interface a new parameter, keepLin, is introduced whenever polarization calibration can be applied to specify that the output is to remain in the linear basis (XX,YY,XY,YX).

IV. Test

A robust test of the keepLin option is to apply polarization calibration to MeerKAT observations of the polarized calibrator 3C138 with and without the keepLin option using Obit task Splat. The resultant datasets were then both imaged in I, Q, U and V using Obit task/MFImage. The two sets of images are visually indistinguishable and a comparison of Faraday rotation fits is given in Figure 1.

National Radio Astronomy Observatory, 520 Edgemont Rd., Charlottesville, VA, 22903 USA email: bcotton@nrao.edu

[1]http://www.cv.nrao.edu/∼bcotton/Obit.html

V. Discussion

The visual comparison of calibrated data and imaging products as well as Figure 1 shows that the changes, allowing polarization calibration to produce data in a linear feed basis, is working as intended. The comparison with the polarization results of [4] in Figure 1 show good agreement with fractional lilnear polarization but some differences in EVPA and rotation measure. The MeerKAT data in this figure has not been corrected for ionospheric Faraday rotation but the differences seem large for that. 3C138 has had a recent outburst which has made some change to the EVPA (R. Perley private communication).

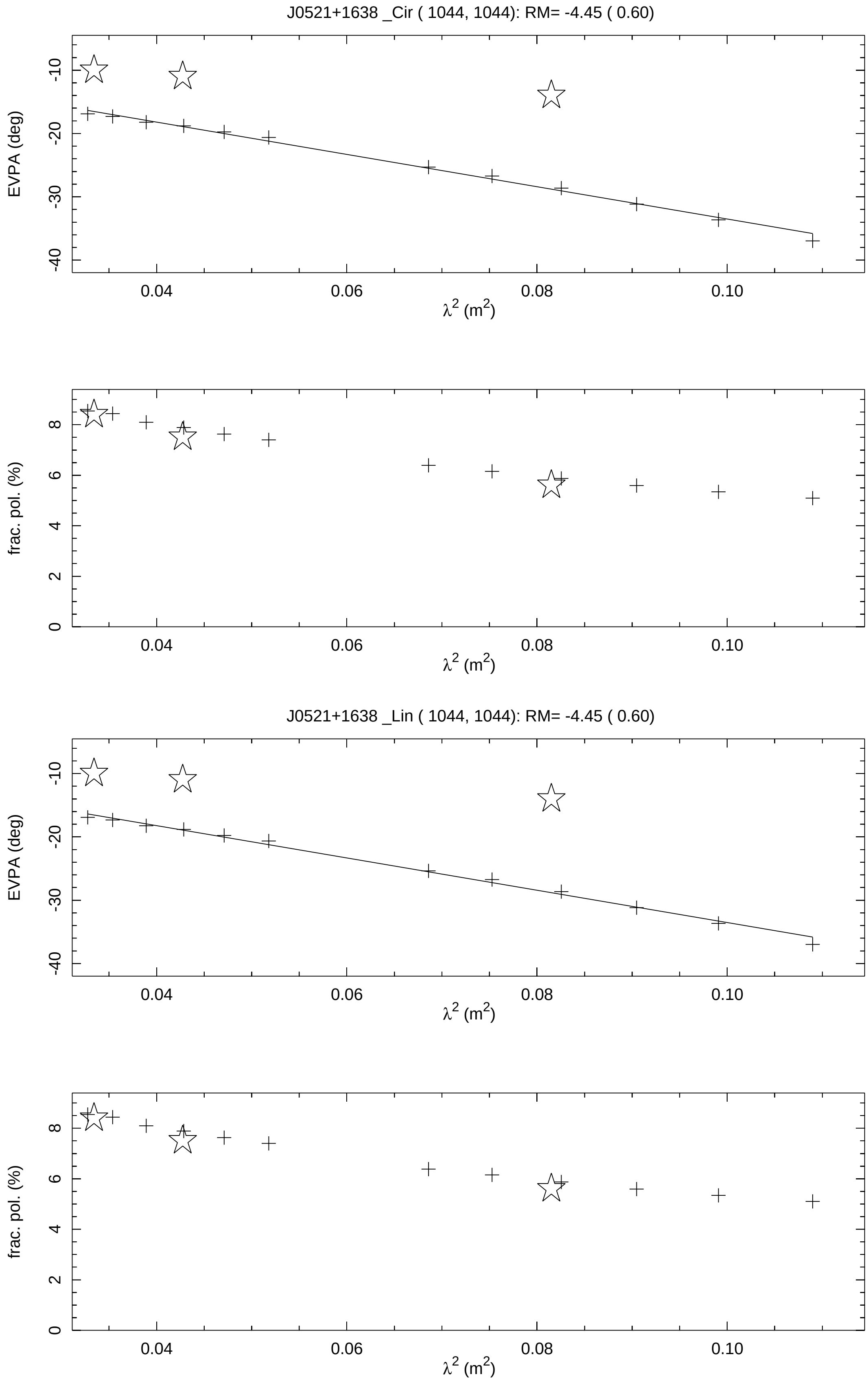

Fig. 1. Polarization of 3C138.
Top: From imaging using data in which the output of applying polarization calibration was in the circular basis. The upper panel gives the EVPA as a function of wavelength^2 where the pluses are the subband averages and the line gives the fitted rotation measure. Stars show the values from [4].
Bottom: Like **Top** but the the output of applying polarization calibration was in the linear basis.

APPENDIX

The test of ObitUVCalPolarization class function Cir2Lin is given in Figure 2. The routine to compute the inverse prefect circular feed, ObitMatx:ObitMatxIPerfCirJones is shown in Figure 3.

REFERENCES

[1] W. D. Cotton, "Obit: A Development Environment for Astronomical Algorithms," *PASP*, vol. 120, pp. 439–448, 2008.

[2] ——, "Onaxis Instrumental Polarization Calibration for Circular Feeds," *Obit Development Memo Series*, vol. 30, pp. 1–14, 2012. [Online]. Available: https://www.cv.nrao.edu/∼bcotton/ObitDoc/PCalRL.pdf

[3] ——, "Onaxis Instrumental Polarization Calibration for Linear Feeds," *Obit Development Memo Series*, vol. 32, pp. 1–11, 2015. [Online]. Available: https://www.cv.nrao.edu/∼bcotton/ObitDoc/

[4] R. A. Perley and B. J. Butler, "Integrated Polarization Properties of 3C48, 3C138, 3C147 and 3C286," *ApJS*, vol. 206, p. 16, 2013.

```
/**
 * Convert circular (RR,LL..) to linear (XX,YY...)
 * \param in     Calibration Object.
 * \param time  Time of datum
 * \param ant1  first antenna number of baseline
 * \param ant2  second antanna of baseline.
 * \param RP     Random parameters array.
 * \param visIn  input/output visibility as an array of floats
 */
static void
Cir2Lin (ObitUVCal *in, float time, olong ant1, olong ant2, ofloat *RP, ofloat *visIn)
{
  gboolean flag;
  ObitUVDesc *desc = in->myDesc;
  ObitUVCalPolarizationS *me = in->polnCal;
  ObitMatx *Jones = NULL;
  olong j, incs=desc->incs;
  olong ndim=2, naxis[]={4,1};


  /* Need rotation matrix? Work arrays?*/
  if (me->C2L_Matrix==NULL) {
    naxis[1] = 4;
    me->C2L_Matrix = ObitMatxCreate(OBIT_Complex, ndim, naxis);
    naxis[1] = 1;
    if (in->workMatx1==NULL) in->workMatx1 = ObitMatxCreate(OBIT_Complex, ndim, naxis);
    if (in->workMatx2==NULL) in->workMatx2 = ObitMatxCreate(OBIT_Complex, ndim, naxis);
    Jones = ObitMatxCreate(OBIT_Complex, ndim, naxis);
    ObitMatxIPerfCirJones(Jones);  /* Inverse perfect feed */
    ObitMatxOuterMult2C(Jones, Jones, me->C2L_Matrix);
    Jones =  ObitMatxUnref(Jones); /* cleanup */
  } /* end create rotation matrix */

 /* Convert vis to linear - assume weights are OK and are from input linear */
 /* anything flagged? -> zero all */
 flag = FALSE;  for (j=0; j<12; j+=3) if (visIn[j+2]<=0.0) {flag = TRUE; break;}
 if (flag) {
   for (j=0; j<12; j++) visIn[j] = 0.0;
   return;
 }
 /* Load input vector [RR,RL,LR,LL]*/
 ObitMatxSet(in->workMatx1, (void*)&visIn[0],      0, 0);
 ObitMatxSet(in->workMatx1, (void*)&visIn[2*incs], 1, 0);
 ObitMatxSet(in->workMatx1, (void*)&visIn[3*incs], 2, 0);
 ObitMatxSet(in->workMatx1, (void*)&visIn[incs],   3, 0);

 /* Multiply by inverse Muller matrix */
 ObitMatxVec4Mult(me->C2L_Matrix, in->workMatx1, in->workMatx2);

 /* unload output vector from [XX,YY,XY,YX] */
 ObitMatxGet(in->workMatx2, 0, 0, (void*)&visIn[0]);
 ObitMatxGet(in->workMatx2, 3, 0, (void*)&visIn[incs]);
 ObitMatxGet(in->workMatx2, 1, 0, (void*)&visIn[2*incs]);
 ObitMatxGet(in->workMatx2, 2, 0, (void*)&visIn[3*incs]);

} /* end Cir2Lin */
```

Fig. 2. ObitUVCalPolarization class function Cir2Lin to convert circular basis data to linear.

```
/** Inverse perfect circular feed Jones matrix
    Really just the perfect linear Jones matrix */
void ObitMatxIPerfCirJones(ObitMatx *out)
{
  ofloat elp_x=0.0, elp_y=0.0, ori_x=0.0, ori_y=G_PI*0.5;
  ofloat angle[4], sina[4], cosa[4], Jones[8];

  angle[0] = G_PI*0.25+elp_x; angle[1] = G_PI*0.25-elp_y;
  angle[2] = ori_x;           angle[3] = ori_y;
  ObitSinCosVec(4, angle, sina, cosa);

  Jones[0] =  cosa[0]*cosa[2]; Jones[1] = -cosa[0]*sina[2];
  Jones[2] =  sina[0]*cosa[2]; Jones[3] =  sina[0]*sina[2];
  Jones[4] =  sina[1]*cosa[3]; Jones[5] = -sina[1]*sina[3];
  Jones[6] =  cosa[1]*cosa[3]; Jones[7] =  cosa[1]*sina[3];

  /* Matrix out */
  COMPLEX_SET (out->cpx[0], Jones[0], Jones[1]);
  COMPLEX_SET (out->cpx[1], Jones[2], Jones[3]);
  COMPLEX_SET (out->cpx[2], Jones[4], Jones[5]);
  COMPLEX_SET (out->cpx[3], Jones[6], Jones[7]);
} /* end ObitMatxIPerfCirJones */
```

Fig. 3. ObitMatx class function ObitMatxIPerfCirJones for the computation of the inverse Jones matrix for perfect Circular feeds.