EVLA Polarizaton Calibration in Low SNR

W. D. Cotton (NRAO), September 3, 2024

***Abstract*—A image based calibration scheme for linear polarization angle measurements is presented. Imaging uses the known position of the calibrator(s) to greatly reduce the contributions of the other sources of emission in the field of view which are in visibility data at full strength. It is successfully applied in a low Signal–to–Noise Ratio case where a more traditional visibility based scheme failed.**

***Index Terms*—Polarization Calibration**

I. INTRODUCTION

POLARIZATION calibration of radio interferometric data is relatively straightforward in high signal-to-noise cases in which the polarized signal from the calibration source(s) dominates the instrumental contribution which, in turn, dominates other sources of noise. In the low signal-to-noise regime, polarization calibration can be challenging. The issues can be more problematic at low frequencies where the standard polarized calibrators are weakly polarized, strong RFI is common and solar activity can disturb the ionosphere. This memo describes the analysis of a technique for assisting the calibration of linear polarization angle in the Obit package [1][1] .

II. POLARIZATION CALIBRATION BASICS

Radio interferometers usually measure source polarization using two sets of detectors (“feeds”) which measure two orthogonal polarization states of the incoming signal. These are usually nominally right and left circular or two linear polarizations. In practice, the feeds are imperfect and actually measure two, nearly orthogonal, elliptical polarization states. There are two distinct aspects of calibration of the polarization response of radio interferometers, 1) determining and removing the spurious polarized response to unpolarized emissions and 2) the calibration of the delay and phase difference between the two sets of polarized detectors. The spurious polarized response is the result of the difference between the nominal feed polarizations and their actual ones. For antennas with circularly polarized feeds, the Right-Left (R-L) phase difference error is also the error of the derived angle of the polarized emission on the sky. For arrays with linear feeds, the “X-Y” phase error rotates linear polarization into circular. The following describes the situation for circular feed systems such as the VLA cassegrain frequencies.

A. Instrumental Polarization

Determination of the spurious polarized response to an unpolarized source is simplified if unpolarized sources are available; the response is, by definition, all instrumental. Fortunately, at low frequencies, good unpolarized calibrators are abundant.

There are two commonly used parameterizations of instrumental polarization; the “D terms” and ellipticity-orientation. The “D terms” are the (small) complex fraction of the orthogonal polarization that is added to the desired polarization. Ellipticity and orientation is the description of the polarization state that each of the feeds is sensitive to. These two are mathematically equivalent [2]. The Obit implementation uses ellipticity-orientation; for circular feeds this is described in [3] and for linear feeds in [4].

B. R-L Phase

The usual parallel hand calibration technique is to make separate solutions for the system of baselines using each of the two orthogonal hands. This leaves an arbitrary phase and delay difference between the two systems. If the same “reference antenna” is used for both parallel systems, the difference between the two is that of the reference antenna. For data in a circular feed basis, an error in the R-L phase difference results in a corresponding error in the Electric Vector Polarization Angle (EVPA). Determining the R-L phase offset requires either hardware calibration or observations of a polarized celestial source. The EVLA has no hardware calibration.

C. Standard Obit Calibration

The standard Obit EVLA calibration pipeline [5] uses three steps:

1) **R-L Delay Calibration:** Task RLDly is used to determine the R-L (X-Y) delay at the Spectral Window level given a polarized calibrator and (optionally) a polarization model. RLDly is described in detail in [6].
2) **Instrumental Polarization Calibration:** Task PCal is used to determine the instrumental parameters in blocks of channels from one or more calibrators. If the calibrator has a known state (either polarized or unpolarized) this information can be provided. Unknown calibrators observed over a range of paralactic angle can have their polarization states solved for. Depending on the calibrators and parameters provided, the instrumental polarization, unknown source polarizations and the R-L phase can be solved for.
3) **R-L Bandpass Calibration:** Task RLPass can be used to tweak the R-L phases in blocks of channels given a polarized source and a polarization model

III. OBIT LOW SNR MODIFICATION

The low SNR case is especially common at low frequencies (below 5 GHz) where the standard polarized calibrators

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

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

become increasingly depolarized and where RFI can be fierce, especially in the more compact EVLA configurations. The small difference in the fringe rates between target sources and sources of interference on short baselines diminishes the fringe rate smearing reducing the response to RFI on longer baselines. Low frequencies are also sensitive to ionospheric disturbances when the sun is active and short baselines to solar emission.

The two basic components of polarization calibration, instrumental parameters and R-L phase and delay errors must still be determined and removed. At lower frequencies ionospheric effects may become important.

1) **Instrumental Polarization Calibration:** This is best done using unpolarized calibrators. Since there is no source polarization to have an R-L phase/delay difference, this can be done first. Multiple sources and scans can be used with only the ellipticities and orientations being solved for using PCal
2) **Image Based R-L Calibration:** Corrections to the EVPA can be derived from an image of a polarized calibrator with adequate spectral resolution and a model of the calibrator, i.e., Rotation Measure(RM) and EVPA at zero wavelength. The correction for each sub band plane is the difference between the EVPA for that plane (0.5*atan2(U,Q) where Q and U are the Stokes parameters at the position of the Stokes I peak) and that derived from the model. For data in a circular basis (RR,LL,RL,LR) the EVPA correction can be applied as a difference in the R and L phase calibrations and is easily implemented as a bandpass table. The Obit python script given in Section VI will determine the EVPA corrections and apply them to an existing bandpass table and write an updated bandpass table. When applied to the data, the derived EVPA should be correct.
3) **Ionospheric Faraday Rotation:** At frequencies below a couple GHz, solar activity can disturb the ionosphere resulting in time and direction dependent Faraday rotation. A discussion of a technique to reduce this effect is given in [7].

IV. EVLA EXAMPLE

The technique described in Section III was applied to an EVLA dataset obtained in October 2023, during a period of enhanced solar activity. The test dataset was observed in S Band (2-4 GHz) in D configuration (the most compact) and includes calibrators 3C48, 3C147 and J0029+3456. 3C147 is completely depolarized in S Band [8]; this reference gives the fractional polarization of 3C48 as rising through 2% so should be detectable. J0029+3456 appears unpolarized.

Strong, persistent interfering signals are common in S Band affecting as much as half of the data. So much of the data is affected that the standard RFI editing procedures have difficulty in distinguishing good data from bad; a heavy handed approach was taken. A dirty (i.e. no CLEANing) image was made using Obit task MFImage [9] with relatively high spectral resolution, maxFBW (maximum fractional bandwidth) of 0.3% which resulted in 240 sub band channels. Each of these had a few spectral channels. A visual inspection of the image was used to identify the ranges of spectral channels degraded by RFI. This RFI appears to be persistent so the affected ranges of channels were flagged for all times. The standard calibration pipeline was rerun with this flagging.

The effects of the standard calibration can be evaluated by examining the EVPA derived from Stokes Q and U images of 3C48 as a function of λ^2 and comparing with the polarization model of [8]. This comparison is given in Figure 1 **Left**. The values obtained from the image are substantially in error, of order 90°. The approach of Section III was adopted.

The instrumental polarization calibration step from Section III was applied using 3C147 and J0029+3456 as unpolarized calibrators generating an AIPS PD table. Even with the instrumental contributions removed, task RLDly was unable to find usable solutions for the R-L delay and phase. The image based R-L calibration from Section III was used instead.

3C48 was imaged using the previous standard pipeline calibration and the AIPS PD instrumental polarizaton table generated in the previous step using a maximum fractional bandpass of 2% (maxFBW=0.02) to generate an image with 37 sub band planes, many are blanked due to RFI. The resultant Q and U images were used with the script given in Section VI to derive a bandpass table with corrections. The polarization model used was RM=-68.0 rad/m^2, EVPA=122° [8]. 3C48 was reimaged using this updated calibration. The comparison of the corrected Q and U images is given in Figure 1 **Right**. Except for the very bottom of S band where 3C48 is strongly depolarized, the values of EVPA scatter about the expected line.

V. DISCUSSION

An alternative approach to EVPA calibration is presented that uses images rather than visibility data directly. Imaging uses the known position of the calibrator to greatly reduce the contributions of other sources of emission in the field which are all represented in the visibility data. This approach appears to work in the test case presented for which a more traditional calibration scheme failed.

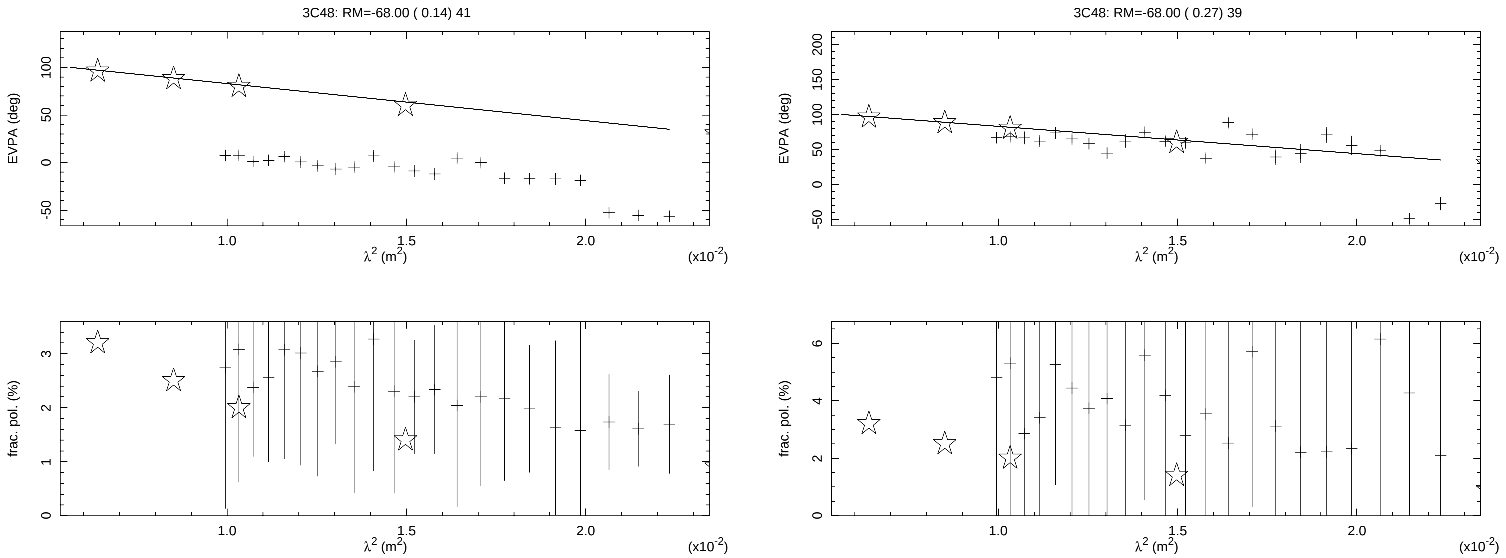

Fig. 1. Subband EVPA and fractional polarization for 3C48. The solid line represents the polarization model of [8] and the stars are the tabulated values from the same reference. Upper panels are EVPA v. λ^2 and the lower panels are fractional polarization. Error bars are included when they exceed the size of the symbol. **Left:** Standard calibration. **Right:** After R-L recalibration.

VI. APPENDIX: FIXEVPA.PY

The text of the Obit python script FixEVPA is shown in Figures VI - VI. This script takes spectrally resolved Q and U FITS images from imager MFImage and a bandpass (AIPS BP) table on an AIPS UVTAB FITS format data set to create an updated bandpass table with corrections to the EVPA. This implementation needs uv data in a circular basis (RR,LL,RL,LR). This script is distributed as $OBIT/share/scripts/FixEVPA.py.

VII. ACKNOWLEDGMENT

I would like to thank Fatemeh Tabatabael and Amir Yarahmadi for access to their calibration data.

REFERENCES

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

[2] A. R. Thompson, J. M. Moran, and G. W. Swenson, Jr., *Interferometry and Synthesis in Radio Astronomy, 2nd Edition*, Thompson, A. R., Moran, J. M., & Swenson, G. W., Jr., Ed. Wiley-Interscience, 2001.

[3] W. D. Cotton, "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

[4] ——, "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/

[5] ——, "EVLA Continuum Scripts: Outline of Data Reduction and Heuristics," *Obit Development Memo Series*, vol. 29, pp. 1–29, 2019. [Online]. Available: https://www.cv.nrao.edu/∼bcotton/ObitDoc/EVLAObitScripts.pdf

[6] ——, "a New Method for Cross Polarized Delay Calibration of Radio Interferometers," *Journal of Astronomical Instrumentation*, vol. 1, no. 1, p. 1250001, Dec. 2012.

[7] ——, "Ionospheric Faraday Rotation Correction," *Obit Development Memo Series*, vol. 83, p. 1, 2024. [Online]. Available: https://www.cv.nrao.edu/∼bcotton/ObitDoc/IonTECor.pdf

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

[9] W. D. Cotton, J. J. Condon, K. I. Kellermann, M. Lacy, R. A. Perley, A. M. Matthews, T. Vernstrom, D. Scott, and J. V. Wall, "“The Angular Size Distribution of μjy Radio S”ources”," *ApJ*, vol. 856, p. 67, 2018.

Fig. 2. Obit python script to determine EVPA corrections and apply them to a bandpass (AIPS BP) table.

```
# Generate a BP table to correct errors in EVPA derived from a scan on 3C48
# Works on data in a circular basis
# Needs:
# - Q and U CLEAN FITS images of 3C48 with enhanced spectral resolution
#   derived applying a previously applied, but inadequate, polarization
#   calibration.
# - Partially poln calibrated UV data set in AIPS FITAB format with a
#   BP table to which corrections can be applied.
# Execute as
#exec(open('FixEVPA.py').read())
import UV, Image, OErr, FArray, ImageDesc, UVPolnUtil, History
import math
err = OErr.OErr()
FBlank = FArray.fblank  # Blanked pixel value

# UV data file name in cwd
uv_file='My_Data.uvtab'
BPinVer  = 2  # Input BP table version
BPoutVer = 3  # Output BP table version

# 3C48 Q&U images
Q_file='3C48_QPol.fits'
U_file='3C48_UPol.fits'

# 3C48 Polarization model from Perley & Butler 2013, ApJS, 206,16
cal = '3C48'
EVPA_cal = math.radians(122.)  # EVPA (deg) @ 0 wavelength
RM_cal   = -68.                # Rotation measure (rad/m^2)
ra  = ImageDesc.PHMS2RA("01:37:41.299")    # Calibrator RA
dec = ImageDesc.PDMS2Dec("33:09:35.133")   # Calibrator Dec

# Shouldn't need to edit below here
# files to Obit objects
uv = UV.newPFUV("in UV", uv_file, 0, True, err)
# Open and close to check
uv.Open(UV.READONLY, err); uv.Close(err);
uv_desc = uv.Desc.Dict   # UV data header as python dict
bpin  = uv.NewTable(Table.READONLY, "AIPS BP", BPinVer, err)
numPol = min(2, uv_desc['inaxes'][uv_desc['jlocs']])
numIF = uv_desc['inaxes'][uv_desc['jlocif']]
numChan=uv_desc['inaxes'][uv_desc['jlocf']]
bpout = uv.NewTable(Table.WRITEONLY, "AIPS BP", BPoutVer, err,
                    numPol=numPol, numIF=numIF, numChan=numChan)
bpout = Table.PClone(bpin, bpout)  # Clone input table
OErr.printErrMsg(err,message='Error with uv data')  # Error check

# Images
im_Q = Image.newPFImage("Q", Q_file, 0, True, err)
im_U = Image.newPFImage("U", U_file, 0, True, err)
# Open and close to check
im_Q.Open(Image.READONLY, err); im_Q.Close(err);
im_U.Open(Image.READONLY, err); im_U.Close(err);
OErr.printErrMsg(err,message='Error with image')  # Error check
Q_desc = im_Q.Desc.Dict       # Q image header
Q_info = im_Q.Desc.List.Dict  # List of frequency bins
nterm = Q_info['NTERM'][2][0]  # no. spectral terms in images
nspec = Q_info['NSPEC'][2][0]  # Number of spectral planes
```

Fig. 3. Obit python script continued.

```
# Pixel number of position of calibrator
pix = ImageDesc.PGetPixel(im_Q.Desc, [ra,dec], err)
pixel = [int(pix[0]+0.5), int(pix[1]+0.5)]  # Round to nearest pixel

# Get list of subband bins as lambda^2,
# low, center, high (freq), Q, U, corr
clight = 2.997924562e8   # Speed of light m/s
bins = []
for i in range(0,nspec):
    key = "FREL%4.4d"%(i+1);  lof = Q_info[key][2][0]
    key = "FREQ%4.4d"%(i+1); cenf = Q_info[key][2][0]
    key = "FREH%4.4d"%(i+1);  hif = Q_info[key][2][0]
    # center to wavelength^2
    cenl2 = (clight/cenf)**2
    bins.append([lof,cenl2,hif, FBlank, FBlank, FBlank])

# end loop

# Get subband flux densities at Q,U position of peak
for i in range(0,nspec):
    ip = i+nterm+1  # 1-rel Plane number in cube
    im_Q.GetPlane(None, [ip,1,1,1,1], err)
    Q_flux = im_Q.FArray.get(pixel[0], pixel[1])
    im_U.GetPlane(None, [ip,1,1,1,1], err)
    U_flux = im_U.FArray.get(pixel[0], pixel[1])
    # Valid data?
    if (im_Q.FArray.RMS!=0.0) and (im_U.FArray.RMS!=0.0):
        bins[i][3] = Q_flux; bins[i][4] = U_flux
        # Get correction (rad)
        evpa_obs = 0.5*math.atan2(U_flux,Q_flux)
        evpa_mod = EVPA_cal + bins[i][1]*RM_cal
        bins[i][5] = -2*(evpa_obs-evpa_mod)  # Correction
        #print (i, "correction=",math.degrees(bins[i][5]), bins[i][1]) # debug
# End loop
OErr.printErrMsg(err,message='Error reading images')  # Error check

# Fill in missing values, first valid
last_corr = FBlank
for i in range(0,nspec):
    if bins[i][5]!=FBlank:
        last_corr = bins[i][5]
        break

# Fill in with last_corr
for i in range(0,nspec):
    if bins[i][5]==FBlank:
        bins[i][5] = last_corr
    else:
        last_corr = bins[i][5]

# List of Frequencies per IF/channel
freqs = UVPolnUtil.GetFreqArr(uv, err)
OErr.printErrMsg(err,message='Error reading frequencies')  # Error check
```

Fig. 4. Obit python script continued.

```
# Get list of corrections (rlcorr) per channel
nch = len(freqs); last_bin = 0
rlcorr_c = nch*[0.0]; rlcorr_s = nch*[0.0]
for i in range(0,nch):
    f = freqs[i]
    if f<bins[last_bin][0]:
        # in an earlier bin, search from start
        last_bin = 0
    # Search if needed
    while not ((f>=bins[last_bin][0]) and (f<=bins[last_bin][2])):
        last_bin += 1
    rlcorr_c[i] = math.cos(bins[last_bin][5]);
    rlcorr_s[i] = math.sin(bins[last_bin][5]);

# Adjust BP table BPinVer to BPoutVer
bpin.Open(Table.READONLY, err); bpout.Open(Table.WRITEONLY,err)
nrow = bpin.Desc.Dict[’nrow’]
for irow in range(1, nrow+1):
    # Update row REAL 2 and IMAG 2 (R-L phase)
    row=bpin.ReadRow(irow,err)
    for i in range(0,nch):
        if (row[’REAL 2’][i]!=FBlank):
            g_in  = complex(row[’REAL 2’][i], row[’IMAG 2’][i]);
            g_add = complex(rlcorr_c[i],rlcorr_s[i])
            g_out = g_in * g_add   # Update
            row[’REAL 2’][i] = g_out.real; row[’IMAG 2’][i] = g_out.imag;
        # End if input valid
    # End loop over channels
    bpout.WriteRow(irow,row,err)
# End loop over rows

bpin.Close(err); bpout.Close(err)
OErr.printErrMsg(err,message=’Error updating BP table’)  # Error check

# Add history
pgm = "FixEVPA"
outHistory = History.History("history", uv.List, err)
outHistory.Open(History.READWRITE, err)
outHistory.TimeStamp(" Start Obit "+pgm,err)
outHistory.WriteRec(-1,pgm+" BPIn   = "+str(BPinVer),err)
outHistory.WriteRec(-1,pgm+" BPOut  = "+str(BPoutVer),err)
outHistory.WriteRec(-1,pgm+" Q_file = "+Q_file,err)
outHistory.WriteRec(-1,pgm+" U_file = "+U_file,err)
outHistory.WriteRec(-1,pgm+" cal    = "+cal,err)
outHistory.WriteRec(-1,pgm+" EVPA_cal = "+str(math.degrees(EVPA_cal)),err)
outHistory.WriteRec(-1,pgm+" RM_cal   = "+str(RM_cal),err)
outHistory.Close(err)
```