Convention for UVW calculations in CASA

Urvashi Rau

February 8, 2013

Abstract

This note describes the UVW coordinate system convention defined for Measurement Sets, and compares it with the *uvw* values actually present in our datasets. The MS 2.0 definition corresponds to a right-handed UVW coordinate system and a baseline convention of *ant2-ant1*. The convention present in CASA MSs starts with the above, flips the signs of all three coordinates, and uses a baseline convention of *ant2-ant1*. This violates the documented MS convention, but it makes the baseline UVWs consistent with AIPS which follows a right handed UVW system with a baseline convention of *ant1* - *ant2*. In all cases, *index(ant1)*<*index(ant2)*. The source of the sign flips in the CASA *uvw* calculators has been identified.

Measurement Set Definition : According to the Measurement Set 2.0 documentation, UVW is a right-handed coordinate system, with W pointing towards the source, and a baseline convention of *ant2* - *ant1* where *index(ant1)*<*index(ant2)*.

Consider an XYZ Celestial coordinate system centered at the location of the interferometer, with X towards the East, Z towards the NCP and Y to complete a right-handed system. The UVW coordinate system is then defined by the hour-angle and declination of the phase-reference direction such that (a) when the direction of observation is the NCP (ha=0,dec=90), the UVW coordinates are aligned with XYZ, (b) V, W and the NCP are always on a Great circle, (c) when W is on the local meridian, U points East (d) when the direction of observation is at zero declination, an hour-angle of -6 hours makes W point due East. The l,m,n coordinates are parallel to U,V,W such that l increases with Right-Ascension (or increasing longitude coordinate), m increases with Declination, and n is towards the source. With this convention, images will have Right Ascension increasing from Right to Left, and Declinatiom increasing from Bottom to Top.

Figure 1 shows the positions of four VLA antennas w.r.to geographical East/West/North/South, and the XYZ, UVW and lmn directions for a source source rising in the East at declination +21 deg and hourangle -3.5h (azimuth=91deg,elevation=42deg).

AIPS Definition : UVW is a right-handed coordinate system with W pointing towards the source, and the baseline convention is *ant1-ant2*, where *index(ant1)*<*index(ant2)*. Antenna coordinates are defined in a right-handed XYZ system. The baseline calculations are performed by first taking the difference of the antenna positions in the XYZ system, followed by rotations into UVW.

Note: The AIPS UVW coordinate system is the same as defined for the MS, but the baseline convention is opposite. However, the UVWs calculated within CASA contain a three-way sign flip w.r.to the system documented in the MS 2.0 definition. These sign flips cancel out the difference in baseline convention, and baseline UVW values are consistent between the two packages.

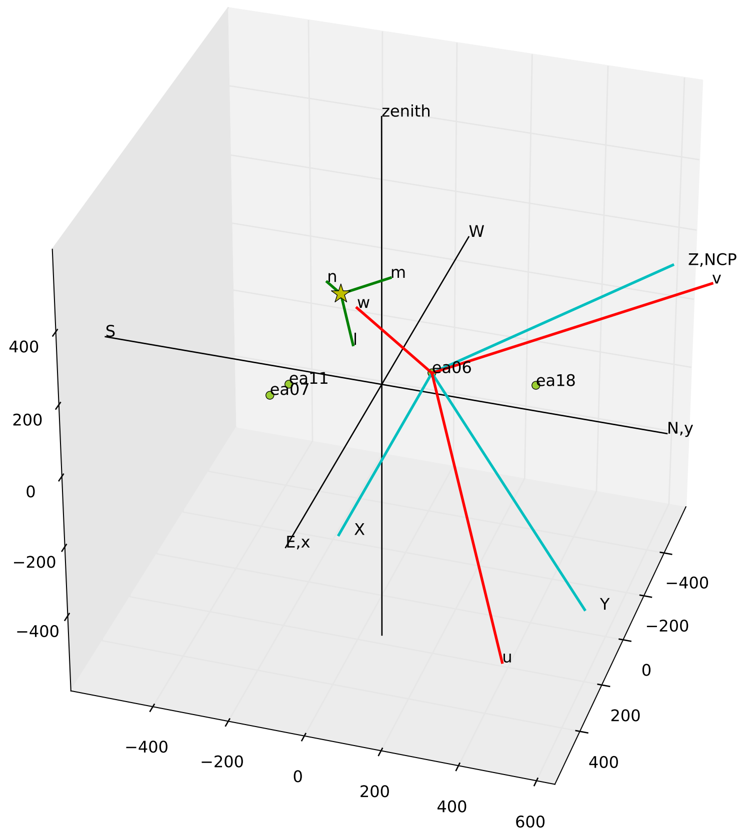


Figure 1: This figure shows the positions of four VLA antennas w.r.to geographical East/West/North/South, the XYZ coordinate system, and the UVW coordinate system for a source rising in the East at declination +21 deg and hourangle -3.49h (az=91deg,el=42deg). The XYZ and UVW coordinates are referenced to the position or antenna ea06.

ID	Ant	Pad	Long.	Lat.	East(m)	North (m)	Elev.
0	ea06	N06	-107.37.06.9	+33.54.10.3	-54.0649	263.8778	-4.2273
1	ea07	E05	-107.36.58.4	+33.53.58.8	164.9788	-92.8032	-2.5268
2	ea11	E04	-107.37.00.8	+33.53.59.7	102.8054	-63.7682	-2.6414
3	ea18	N09	-107.37.07.8	+33.54.19.0	-77.4346	530.6273	-5.5859

The table below lists antenna UVWs calculated in this coordinate system, by projecting antenna vectors onto the UVW axes.

ID	Antenna	u	v	w
0	ea06	0	0	0
1	ea07	-25.2350	-381.2951	170.8261
2	ea11	-50.2824	-337.6665	124.1598
3	ea18	104.7186	245.2779	-24.0005

Antenna UVW values calculated within CASA : Within CASA, the Measures classes implement the coordinate rotations to transform antenna coordinates from XYZ to UVW. For the system shown in Figure 1 (which follows the MS 2.0 definition), our usage of the Measures classes returns antenna UVW values signs flipped on all three axes.

ID	Antenna	u	v	w
0	ea06	0	0	0
1	ea07	25.3069	381.869	-169.525
2	ea11	50.3281	338.088	-122.987
3	ea18	-104.713	-245.366	23.106

A baseline convention of *ant2 - ant1* applied to these values, gives the baseline UVWs present in the UVW column of the MeasurementSet which are also consistent with the values produced within AIPS. The code used to generate the above antenna UVW values is approximately replicated in the Filler, FixVis, Simulator and ShadowFlagger, followed by a consistent use of *ant2-ant1* to calculate baseline UVWs.

Source of the sign flips : All CASA UVW calculators do the opposite of what is documented for Measures/MVBaseline to calculate antenna UVWs w.r.to a reference position.

```
MVBaseline (const MVPosition &pos, const MVPosition &base)
Baseline as difference between positions (first - second (default(0,0,0))
```

All our uses of MVBaseline supply the 'reference position' as the first argument instead of the second.

1. Filler : MVBaseline mvblA(itr->second.getValue(), it->second.getValue()); // itr : ref
2. FixVis : bl_an_p[an] = MVBaseline(refpos_p.getValue(), antPositions_p(an).getValue());
3. Simulator/Flagger : MVBaseline mvblA(obsPos.getValue(), antpos.getValue());

Shadow Flagging in CASA: The sign of 'w' can be used to identify antennas that sit behind others and may be shadowed. In the MS 2.0 definition, if $w < 0$, antenna 2 is behind antenna1.

However, based on the above understanding of the UVW convention present in our MeasurementSets, the value of 'w' (read from the UVW column, or calculated as 'ant2-ant1' using antenna UVW values from the UVW calculators), must be interpreted in a left-handed coordinate system with W pointing away from the source. Therefore, if $w < 0$, antenna1 is behind antenna2.

The shadow-flagging code has been modified[1] in casapy r22765 (casapy 4.1 onwards) to conform to this flipped coordinate system. *Any future change to our usage convention of Measures/MVBaseline, must be accompanied by a change in shadow-flagging logic.*

Figures 2,3 and 4 depict shadow-flagging results on plot of antenna positions, with circles drawn around them. The radii of these circles represent the fraction of data flagged. Also shown, are 3D plots of the antenna positions, the local geographic xyz coordinates (Y=North), the Earth/Celestial-frame XYZ coordinates (Z=NCP), and the UVW coordinates for the current source direction (W=source). From the labeled source direction, one can easily see which antennas should be flagged as being shadowed and confirm from the shadow-flag plots that this is indeed the case.

[1]Before the refactoring of the flagger in early 2011, the shadowing logic in the old flagdata (casapy 3.4 and before) followed a right handed system with 'w' pointing up (same as MS 2.0, opposite to the data), but the antenna convention was $ant1 > ant2$ (opposite to MS 2.0), and baseline convention was ant2-ant1 (same as MS 2.0). Therefore, two minus signs canceled each other, and the old shadow flagging identified shadowed antennas correctly. After the refactoring (casapy 4.0), the logic followed the MS 2.0 definition and shadowing antennas were mis-identified as being shadowed.

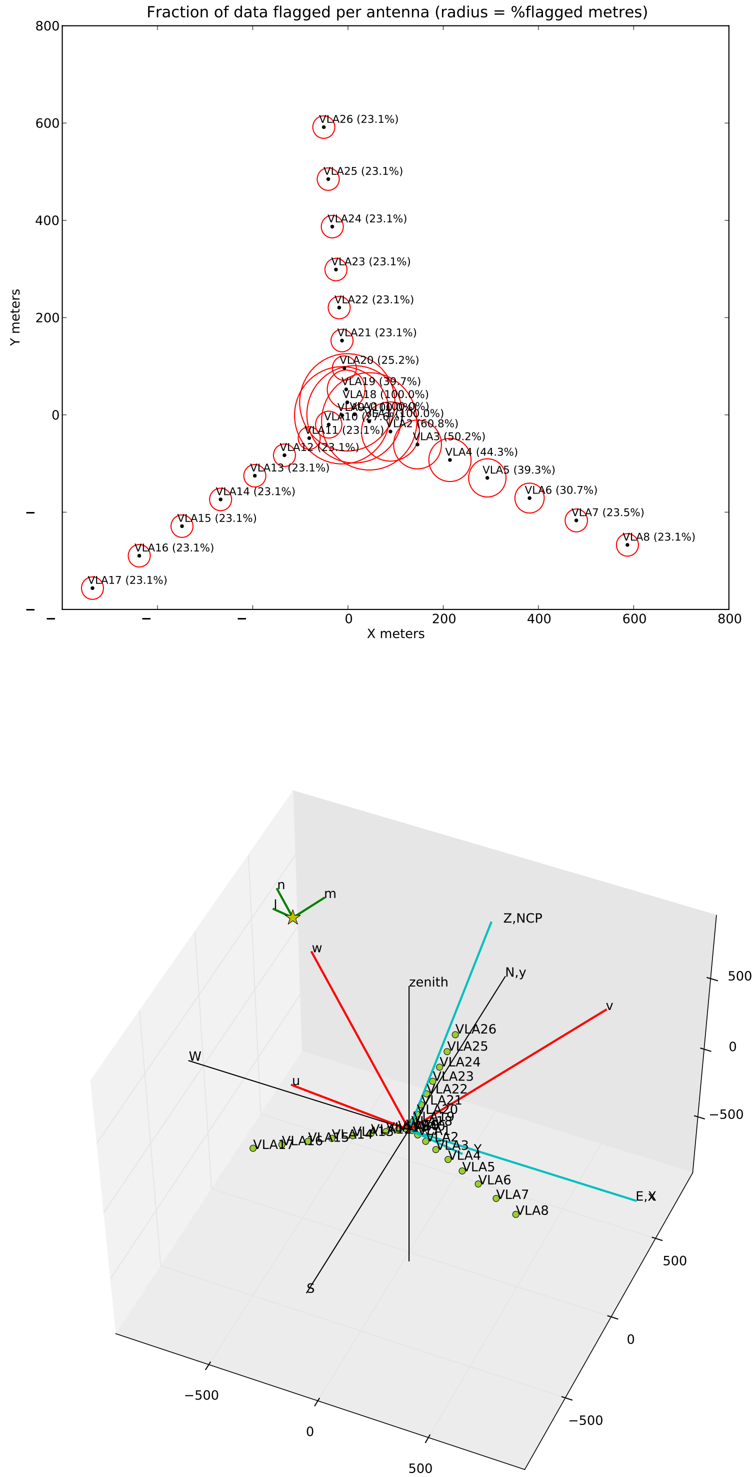


Figure 2: A simulated EVLA dataset : The source is setting towards the NorthWest-West, at HA=+8.0h, DEC=+40.6deg, EL=5deg, AZ=-42deg. Therefore, antennas on the East arm of the VLA are shadowed more than the others. A tolerance of -10.0m was used for shadow-flagging.

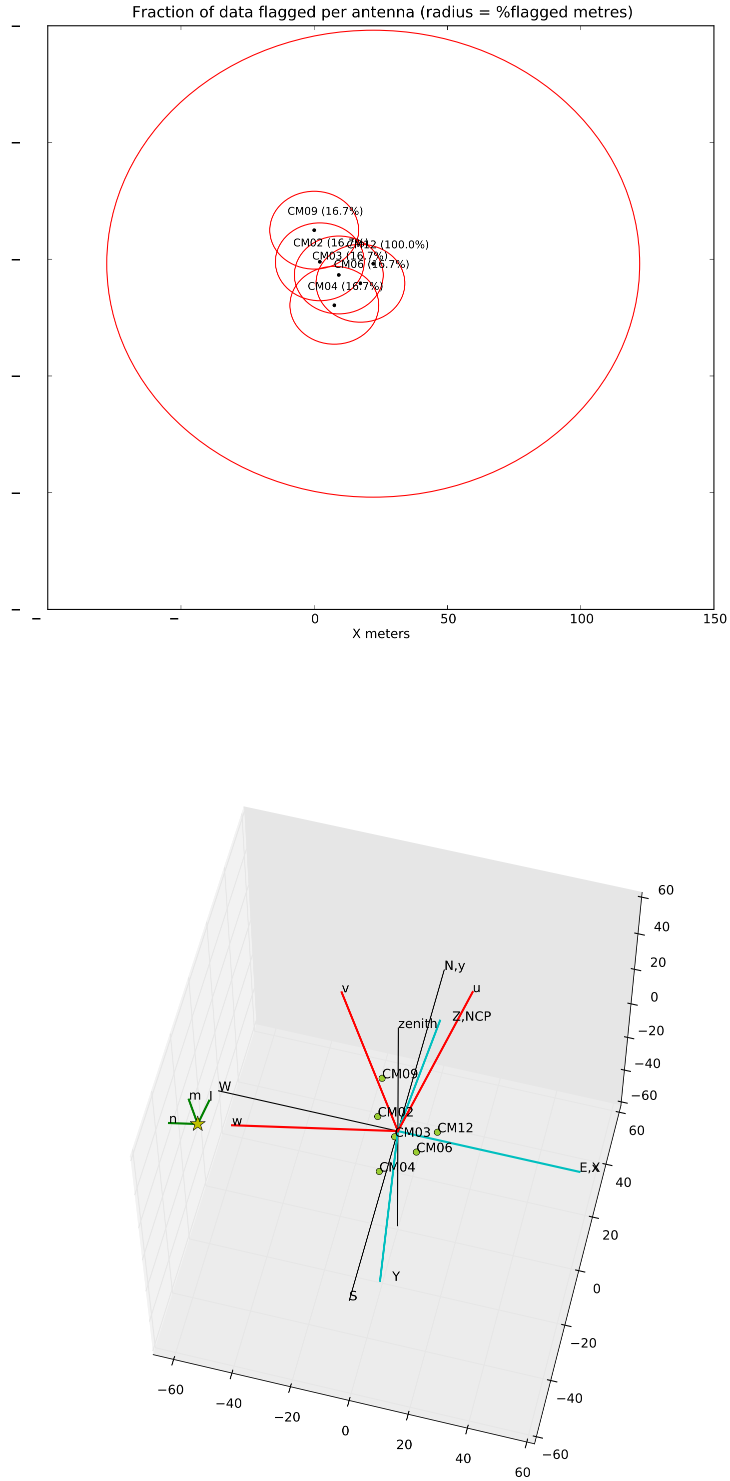
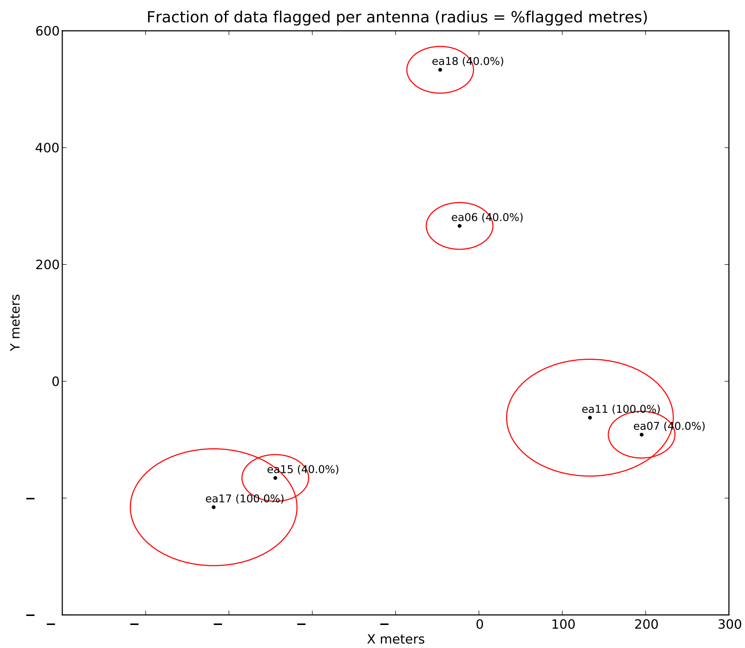


Figure 3: An ALMA dataset : The source is setting towards the SouthWest-West, at HA=+4.82h, DEC=-34.42deg, EL=26.8deg, AZ=-119deg. Therefore, antenna CM12 on the North-East side of all the other antennas, is shadowed the most. A tolerance of 0.5m was used for shadow-flagging.



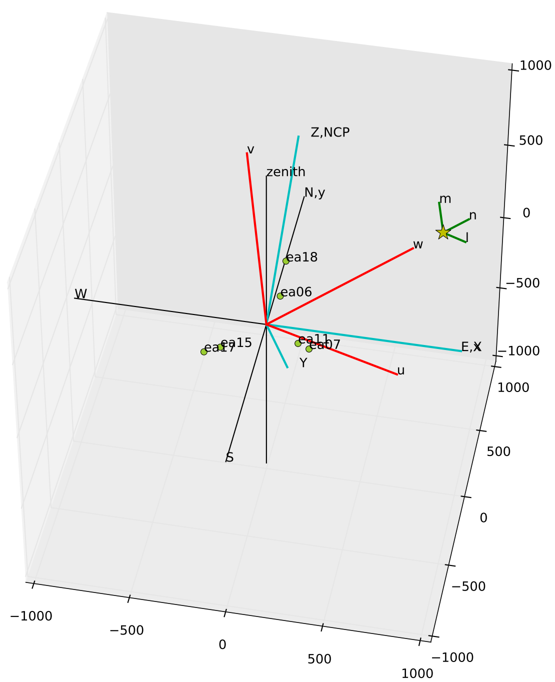


Figure 4: A VLA dataset : The source is rising at almost due East, at HA=-3.5h, DEC=+21.0deg, EL=42.5deg, AZ=91.5deg. Therefore, antennas to the West are more shadowed than others. A tolerance of -60m was used for shadow-flagging (only for demonstration purposes).

Appendix 1 : The following code segment is approximately replicated in the Filler, Simulator, Fixvis and Flagger code in CASA. It uses casacore::Measures to compute antenna UVWs, and is followed by a baseline *uvw* calculation as $uvwAnt[j] - uvwAnt[i]$ where $i < j$.

```
Double Time = vb.timeCentroid()(0);
MEpoch epoch(Quantity((Time), "s"), MEpoch::UT1);
MDirection refdir(vb.phaseCenter());
MPosition obsPos( antennaPositions_p[0] );
MVPosition basePos=obsPos.getValue();
MeasFrame measFrame(obsPos);
measFrame.set(epoch);
measFrame.set(refdir);
MVBaseline mvbl;
MBaseline basMeas;
MBaseline::Ref basref(MBaseline::ITRF, measFrame);
basMeas.set(mvbl, basref);
basMeas.getRefPtr()->set(measFrame);
Int nAnt = antennaPositions_p.nelements();
uvwAnt_p.resize(3,nAnt);
MBaseline::Convert elconv(basMeas, MBaseline::Ref(MBaseline::J2000));
Muvw::Convert uvwconv(Muvw(), Muvw::Ref(Muvw::J2000, measFrame));

for(Int k=0; k< nAnt; ++k)
  {
    MPosition antpos=antennaPositions_p(k);
    MVBaseline mvblA(obsPos.getValue(), antpos.getValue());   <- Source of sign flip.
    basMeas.set(mvblA, basref);
    MBaseline bas2000 =  elconv(basMeas);
    MVuvw uvw2000 (bas2000.getValue(), refdir.getValue());
    const Vector<double>& xyz = uvw2000.getValue();
    uvwAnt_p.column(k)=xyz;
  }
```

Appendix 2 : The following script computes the local xyz coordinates of a list of antennas, and the XYZ and UVW coordinate axes according to the MS 2.0 definition. Antenna *uvw* values are computed by projecting antenna position vectors (in the xyz system) onto the UVW axes.

```
def cdraw(hourangle=-3.5, declination=+20.0, obslatitude=34.0,antennalist=''):
    """
    hourangle : HA in hours. Zero is the local meridian.-6.0 is rising,+6.0 is setting
    declination : DEC in degrees. 0 is on celestial equator (XY plane), +90 is at NCP.
    obslatitude : Latitude of the Observatory ( VLA : 34.0 )
    antennalist : text file name containing the listobs output for antenna info
    """
```

```
    ## Use the antenna coordinates on the surface of the Earth,
    ## referenced to the array center (listobs output)
    eastlocs, northlocs, elevlocs, antnames = readAntListFile(antennalist)
    eastlocs = eastlocs - np.mean(eastlocs)
    northlocs = northlocs - np.mean(northlocs)
    elevlocs = elevlocs - np.mean(elevlocs)
    ## Assign x->East and x-North. This is the local geographic csys
    Xlocs = eastlocs
    Ylocs = northlocs
    ## Construct UVW axes.
    ## Start with  local xyz
    xdir1 = np.array( [AL,0,0], 'float' )
    ydir1 = np.array( [0,AL,0], 'float' )
    zdir1 = np.array( [0,0,AL], 'float' )
    ## Rotate to get 'z' pointed to where 'w' should be when HA=0, DEC=0
    ## Rotate by HA and DEC in appropriate directions.
    ## Rotate by observatory latitude.
    udir = localxyz2uvw( xdir1, hourangle, declination, latrot )
    vdir = localxyz2uvw( ydir1, hourangle, declination, latrot )
    wdir = localxyz2uvw( zdir1, hourangle, declination, latrot )
    ## Calculate UVWs for all antennas by projecting xyz vectors onto UVW.
    axyz=np.array( [0.0,0.0,0.0], 'float')
    antuvws = np.zeros( (len(Xlocs), 3) ,'float')
    for ant in range(0,len(Xlocs)):
        axyz[0] = Xlocs[ant]
        axyz[1] = Ylocs[ant]
        axyz[2] = elevlocs[ant]
        ## Project onto UVW axes.
        antuvws[ant,0] = np.dot(axyz,udir)/np.linalg.norm(udir)
        antuvws[ant,1] = np.dot(axyz,vdir)/np.linalg.norm(vdir)
        antuvws[ant,2] = np.dot(axyz,wdir)/np.linalg.norm(wdir)
    ## Calculate baseline UVWs
    buvws=[]
    for ant1 in range(0,len(Xlocs)):
        for ant2 in range(ant1+1,len(Xlocs)):
            buvws.append( [ant1, ant2, antuvws[ant2]-antuvws[ant1] ] )
    return antuvws, buvws


def localxyz2uvw( xyz, hourangle, declination, latrot ):
    uvwdir = rotz( rotx( rotx( xyz, 90 ) , -1*declination ) , -1*hourangle*15 )
    uvwdir = rotx( uvwdir, latrot )
    return uvwdir
```

Acknowledgements : Thanks to Eric Greisen for information about the AIPS convention, and to several members of the CASA team for finding code and helping to confirm coordinate conventions.