PolConvert usage guide

Although very uncommon, a station may occasionally observe with linear polarisation receivers instead of the more conventionally used circular pols. This mismatch of polarisations is not ideal, but neither is it fatal since it is possible to reconstruct the circular polarisation data from the linear pol data. Such a conversion can be executed using the *PolConvert* program, written by Iván Marti-Vidal and was installed and modified in the CASAdev computer by Mark Kettenis. PolConvert runs in CASA. There are currently two versions of *PolConvert*, these are:
`PolConvert_transfer_EVN.py` - which takes solutions derived from a multi-band data set (ex. continuum pass) and applies the correction to a single band data set (ex. line pass).
`PolConvert_calibrate_EVN.py` - which derives corrections from a single data set (either multi-/single-band) and applies corrections to itself only.

Preparation

Both versions of PolConvert reside in the CASAdev computer. Begin by logging in to CASAdev and initialising CASA:

```
ssh -X jops@casadev
pw: same as all the other computers
. ~/casainit.sh
```

From this point, you should make a folder and copy in your FITS IDI data.

```
mkdir <mydir>
cd <mydir>
scp eee.nfra.nl:/<mypath>/*IDI* .
```

PolConvert is controled by an input file which contains some parameters that must be set by the user. Since each usage of PolConvert will be unique you will need to copy the input file to your directory where you can edit and run it. A copy of the current versions of `PolConvert_calibrate_EVN.py` and `PolConvert_transfer_EVN.py` can be found in `/scratch/ross`.

Below are the first 15 lines of an example input file for `PolConvert_calibrate_EVN.py`. A general explanation is given further below.

```
REFANT = 4 # Antenna to which refer the conversion gain solution (O8)
LINANT = 2 # Antenna with linear feed (EF)
#
REF_IDI = 'eo014_2_1.IDI42' # IDI with calibrator scan
CALRANGE = [0,23,32,39,0,23,39,45] # time range of calibrator scan (J0927+3902)
#
NCHAN = 2048 # Number of channels per IF (to compute the multi-band delay)
NIF = 1  # Number of IFs.
NITER = 2  # Number of X-Y phase-estimate iterations (just 1 should suffice)
#
# List with the names of all FITS-IDIs to convert:
ALL_IDIs = ['eo014_2_1.IDI'+str(i) for i in range(1,100)]
#
# Add this suffix to the converted IDIs (empty string -> overwrite!)
SUFFIX = '.POLCONVERT'
```

The first lines of the input are exemplified above in which Effelsberg (Ef) data were converted from linear to circular polarisation using Onsala (O8) as a circular-pol reference station. The PolConvert uses a basic fringe fit which is performed on a scan of a (preferably) bright source, in this example case the $\sim$ 7 min scan on J0927+3902 was taken from the 42nd FITS IDI file (eo014_2_1.IDI42), you may need to load the data into AIPS to identify which IDI file contains a scan on a bright source - this is simple if the experiment scheduled a fringe finder at the start/end of the observation. After deriving the PolConvert corrections (amplitudes and phases) the other FITS IDI files are processed as specified in the `ALL_IDIs` parameter. The final `PolConverted` FITS IDIs are given a suffix as defined by the `SUFFIX` parameter - note that if you leave this parameter blank you will overwrite the original data, be careful and have fun.

Once these parameters are correctly set PolConvert is run using:

```
casa -c PolConvert_calibrate_EVN.py
or
casa -c PolConvert_transfer_EVN.py
```