Cross Phase Calculation Memo

Ruby Byrne

April 2018

1 Background

One degree of freedom in polarized calibration is the overall phase offset between the x- and y-dipoles. This phase offset, called ϕ for the purposes of this memo, corresponds to a mixing between the Stokes U and V modes. As we do not expect sky emission to be circularly polarized, we can constrain ϕ by minimizing Stokes V emission.

We use pseudo-Stokes parameters to simplify the calculation, allowing us to work in visibility space without gridding. Pseudo-Stokes parameters are calculated from the visibilities as follows:

$$\begin{bmatrix} I^{\text{pseudo}} \\ Q^{\text{pseudo}} \\ U^{\text{pseudo}} \\ V^{\text{pseudo}} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & i & -i \end{bmatrix} \begin{bmatrix} xx^* \\ yy^* \\ xy^* \\ yx^* \end{bmatrix} \tag{1}$$

It is clear that the transformation $x \rightarrow xe^{-i\phi/2}$ and $y \rightarrow ye^{i\phi/2}$ leaves pseudo-I and pseudo-Q unchanged, but affects the pseudo-U and pseudo-V parameters.

2 Calculating the Cross Phase

To calculate ϕ, we minimize the quantity $\sum_n |V_n^{\text{pseudo}}|^2$ where n denotes the visibility index. We write the visibilities in terms of an amplitude and phase, such that $(xy^*)_n = A_n e^{i\delta_n} \rightarrow A_n e^{i\delta_n - i\phi}$ and $(yx^*)_n = B_n e^{i\gamma_n} \rightarrow B_n e^{i\gamma_n + i\phi}$. Now

$$V_n^{\text{pseudo}} = A_n e^{i(\delta_n - \phi + \pi/2)} + B_n e^{i(\gamma_n + \phi - \pi/2)} \tag{2}$$

It follows that

$$|V_n^{\text{pseudo}}|^2 = A_n^2 + B_n^2 - 2A_nB_n \left[\cos(\delta_n - \gamma_n)\cos(2\phi) + \sin(\delta_n - \gamma_n)\sin(2\phi)\right] \tag{3}$$

Now to find ϕ we set $\frac{\partial}{\partial\phi} \sum_n |V_n^{\text{pseudo}}|^2 = 0$. This gives

$$\tan(2\phi) = \frac{\sum_n A_nB_n \sin(\delta_n - \gamma_n)}{\sum_n A_nB_n \cos(\delta_n - \gamma_n)} \tag{4}$$

Rewriting this in terms of the visibilities, we get

$$\tan(2\phi) = \frac{\sum_n \text{Im}\left[(xy^*)_n (yx^*)_n{}^*\right]}{\sum_n \text{Re}\left[(xy^*)_n (yx^*)_n{}^*\right]} \tag{5}$$

3 Implementation in FHD

This calculation is implemented in FHD in the full-pol branch with the function `vis_calibrate_crosspol_phase`.

It is important to note that FHD convention stores visibilities in the `vis_ptr` object, where counterintuitively `vis_ptr[0]` corresponds to yy^*, `vis_ptr[1]` corresponds to xx^*, `vis_ptr[2]` corresponds to yx^*, and `vis_ptr[3]` corresponds to xy^*.

The calculated value of ϕ is stored in the `cal.cross_phase` object and the correct transformation is applied to the antenna gains. Since the raw visibilities are divided by the gains, the transformation $x \to xe^{-i\phi/2}$ is equivalent to $g_x \to g_x e^{i\phi/2}$ and $y \to ye^{i\phi/2}$ is equivalent to $g_y \to g_y e^{-i\phi/2}$. Here g_x is the x-dipole gain (`cal.gain[1]`) and g_y is the y-dipole gain (`cal.gain[0]`).