

pointing_weights

October 8, 2023

1 Pointing Matrix Weights

```
[1]: %matplotlib inline
      %config InlineBackend.figure_format = "retina"
      import matplotlib.pyplot as plt
      import numpy as np
      import seaborn as sns
      sns.set()
      plt.rcParams['figure.figsize'] = (10.0, 6.0)
```

```
[2]: import sympy as sym
      from sympy.vector import CoordSys3D

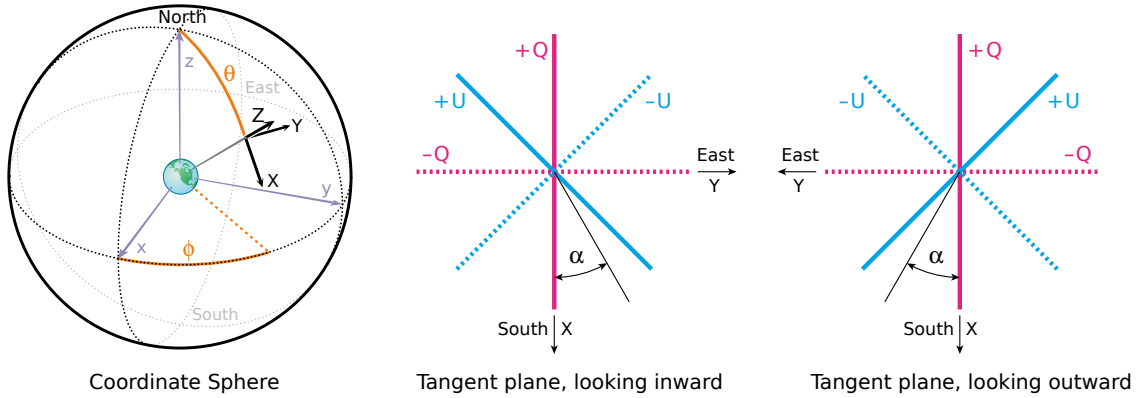
      sym.init_printing()
```

1.1 Background

The “Pointing Matrix” in TOAST is represented by two types of operations. The first is the mapping from geometric detector quaternion pointing to a pixelized representation of the sky. The second is the model of the detector response to incoming polarized light. This model can be expressed as the application of Mueller matrices representing optical elements in the system. The incoming light can be expressed as a vector of Stokes parameters I, Q, U, and V.

The Stokes parameters are defined with respect to the local meridian at the detector line of sight. In the TOAST formalism, the detector frame has the Z axis pointing along the detector line of sight and the X axis aligned with the direction of maximum polarization response. For this exercise, we will use the COSMO convention for the Stokes parameters, since it conveniently also has the Z axis along the line of sight. The final result is easy to swap between COSMO / IAU simply by changing the sign of the U Stokes parameter. In some scenarios, it can be convenient to look at the various angles from the perspective of the instrument “looking out” at the sky (with the detector Z axis going “into the page”). In other cases it makes more sense to visualize the situation “looking in” from the sky. The figures below clearly label which case is being displayed.

COSMO (HEALPix) Conventions



1.1.1 Definitions

We will use α to represent the angle of right-handed rotation from the meridian to the detector polarization orientation (the detector frame X axis) about the line of sight (detector Z axis). We use ω to represent the right-handed angle of rotation, about the line of sight, from the meridian to a direction parallel to the HWP fast axis. The cross-polar leakage of the linearly polarized detector is ϵ . To visually simplify things, we define the polarization coefficient:

$$C = \frac{1 - \epsilon}{1 + \epsilon}$$

And note that there is a factor of 1/2 that can be included in an overall calibration factor in the resulting measured output power.

```
[3]: alpha = sym.Symbol(r"\alpha")
omega = sym.Symbol(r"\omega")
epsilon = sym.Symbol(r"\epsilon")
C = sym.Symbol("C")
alpha, omega, epsilon, C
```

[3]: $(\alpha, \omega, \epsilon, C)$

1.2 Mueller Matrix Representation

Assume we have an input Stokes vector:

$$\vec{S}_{in} = \begin{bmatrix} I_{in} \\ Q_{in} \\ U_{in} \\ V_{in} \end{bmatrix}$$

In the case of no HWP and just a partial linear polarizer followed by a total power measurement, the resulting measured power is:

$$P_{out} = \mathbf{T}_{power} \cdot \mathbf{M}_{det} \vec{S}_{in}$$

Where the total power measurement \mathbf{T} is simply the sum of the top row of the final Mueller matrix. If we introduce a rotating HWP followed by the fixed linear polarizer and then a total power measurement we get:

$$P_{out} = \mathbf{T}_{power} \cdot \mathbf{M}_{det} \cdot \mathbf{M}_{HWP}(t) \vec{S}_{in}$$

The detector and HWP Mueller matrices include a rotational coordinate transform:

$$\mathbf{M}_{HWP} = \mathbf{R}_{HWP}^{-1} \mathbf{M}_{Fixed\ HWP} \mathbf{R}_{HWP}$$

Where the rotation matrix transforms from the measurement frame to the local coordinates of the HWP. Similarly the detector Mueller matrix is an (imperfect) linear polarizer with some rotation from the measurement frame:

$$\mathbf{M}_{det} = \mathbf{R}_{det}^{-1} \mathbf{M}_{Linear\ Pol} \mathbf{R}_{det}$$

1.2.1 HWP Matrix

The ideal, local HWP has Mueller matrix of:

[4]: *# Ideal HWP in local frame:*

```
Lhwp = sym.Matrix([
    [1, 0, 0, 0],
    [0, 1, 0, 0],
    [0, 0, -1, 0],
    [0, 0, 0, -1],
])
Lhwp
```

[4]:
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

and the rotation matrix from the coordinate frame to the local frame is:

[5]:

```
Rhwp = sym.Matrix([
    [1, 0, 0, 0],
    [0, sym.cos(2 * omega), sym.sin(2 * omega), 0],
    [0, -sym.sin(2 * omega), sym.cos(2 * omega), 0],
    [0, 0, 0, 1],
])
Rhwp
```

[5]:
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(2\omega) & \sin(2\omega) & 0 \\ 0 & -\sin(2\omega) & \cos(2\omega) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For simplifying the algebra below, we use $b = 2\omega$. The inverse coordinate transform is a result of swapping ω for $-\omega$.

```
[6]: b = sym.Symbol(r"b")

Rhwp = sym.Matrix([
    [1, 0, 0, 0],
    [0, sym.cos(b), sym.sin(b), 0],
    [0, -sym.sin(b), sym.cos(b), 0],
    [0, 0, 0, 1],
])
Rhwp
```

```
[6]: 
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(b) & \sin(b) & 0 \\ 0 & -\sin(b) & \cos(b) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```

```
[7]: Rhwpinv = sym.Matrix([
    [1, 0, 0, 0],
    [0, sym.cos(b), -sym.sin(b), 0],
    [0, sym.sin(b), sym.cos(b), 0],
    [0, 0, 0, 1],
])
Rhwpinv
```

```
[7]: 
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(b) & -\sin(b) & 0 \\ 0 & \sin(b) & \cos(b) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```

And then the final HWP Mueller matrix becomes:

```
[8]: Mhwp = sym.simplify(Rhwpinv * Lhwp * Rhwp)
Mhwp
```

```
[8]: 
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(2b) & \sin(2b) & 0 \\ 0 & \sin(2b) & -\cos(2b) & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

```

1.2.2 Linearly Polarized Detector

A (partial) linear polarizer has the local Mueller matrix:

```
[9]: Ldet = sym.Matrix([
    [1, C, 0, 0],
    [C, 1, 0, 0],
    [0, 0, 0, 0],
    [0, 0, 0, 0],
])
```

```
])  
Ldet
```

```
[9]:
```

$$\begin{bmatrix} 1 & C & 0 & 0 \\ C & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The Mueller matrix which rotates a Stokes vector in a right-handed counter-clockwise sense is again given by:

```
[10]: Rdet = sym.Matrix([  
    [1, 0, 0, 0],  
    [0, sym.cos(2 * alpha), sym.sin(2 * alpha), 0],  
    [0, -sym.sin(2 * alpha), sym.cos(2 * alpha), 0],  
    [0, 0, 0, 1],  
])  
Rdet
```

```
[10]:
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(2\alpha) & \sin(2\alpha) & 0 \\ 0 & -\sin(2\alpha) & \cos(2\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For simplifying the algebra below, we use $a = 2\alpha$.

```
[11]: a = sym.Symbol(r"a")  
Rdet = sym.Matrix([  
    [1, 0, 0, 0],  
    [0, sym.cos(a), sym.sin(a), 0],  
    [0, -sym.sin(a), sym.cos(a), 0],  
    [0, 0, 0, 1],  
])  
Rdet
```

```
[11]:
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(a) & \sin(a) & 0 \\ 0 & -\sin(a) & \cos(a) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The inverse transform can be obtained by evaluating this at $-\alpha$

```
[12]: Rdetinv = sym.Matrix([  
    [1, 0, 0, 0],  
    [0, sym.cos(a), -sym.sin(a), 0],  
    [0, sym.sin(a), sym.cos(a), 0],  
    [0, 0, 0, 1],  
])  
Rdetinv
```

```
[12]:
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(a) & -\sin(a) & 0 \\ 0 & \sin(a) & \cos(a) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The final detector Mueller matrix in the measurement frame is then:

```
[13]: Mdet = sym.simplify(Rdetinv * Ldet * Rdet)
Mdet
```

```
[13]:
```

$$\begin{bmatrix} 1 & C \cos(a) & C \sin(a) & 0 \\ C \cos(a) & \cos^2(a) & \frac{\sin(2a)}{2} & 0 \\ C \sin(a) & \frac{\sin(2a)}{2} & \sin^2(a) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

1.2.3 Final Expression - Without HWP

The optical response without a HWP is just the linear polarized detector result above. The output Stokes weights are:

```
[14]: s_I = Mdet[0, 0]
s_I
```

```
[14]: 1
```

```
[15]: s_Q = Mdet[0, 1]
s_Q
```

```
[15]: C cos(a)
```

```
[16]: s_U = Mdet[0, 2]
s_U
```

```
[16]: C sin(a)
```

So the final measured power is:

$$P_{out} = \frac{1}{2} \left[I_{in} + \frac{1-\epsilon}{1+\epsilon} [Q_{in} \cos(2\alpha) + U_{in} \sin(2\alpha)] \right] \quad (COSMO)$$

OR

$$P_{out} = \frac{1}{2} \left[I_{in} + \frac{1-\epsilon}{1+\epsilon} [Q_{in} \cos(2\alpha) - U_{in} \sin(2\alpha)] \right] \quad (IAU)$$

1.2.4 Final Expression - Including HWP

The combined optical response of the HWP and detector matrices are then:

```
[17]: Mopt = sym.simplify(Mdet * Mhwp)
Mopt
```

$$[17]: \begin{bmatrix} 1 & C \cos(a-2b) & -C \sin(a-2b) & 0 \\ C \cos(a) & \frac{\cos(2b)}{2} + \frac{\cos(2a-2b)}{2} & \frac{\sin(2b)}{2} - \frac{\sin(2a-2b)}{2} & 0 \\ C \sin(a) & \frac{\sin(2b)}{2} + \frac{\sin(2a-2b)}{2} & -\frac{\cos(2b)}{2} + \frac{\cos(2a-2b)}{2} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

And the output Stokes weights are:

$$[18]: \begin{bmatrix} \mathbf{s_I} = \text{Mopt}[0, 0] \\ \mathbf{s_I} \end{bmatrix}$$

$$[18]: 1$$

$$[19]: \begin{bmatrix} \mathbf{s_Q} = \text{Mopt}[0, 1] \\ \mathbf{s_Q} \end{bmatrix}$$

$$[19]: C \cos(a-2b)$$

$$[20]: \begin{bmatrix} \mathbf{s_U} = \text{Mopt}[0, 2] \\ \mathbf{s_U} \end{bmatrix}$$

$$[20]: -C \sin(a-2b)$$

So the final measured power is:

$$P_{out} = \frac{1}{2} \left[I_{in} + \frac{1-\epsilon}{1+\epsilon} [Q_{in} \cos(2(\alpha-2\omega)) - U_{in} \sin(2(\alpha-2\omega))] \right] \quad (COSMO)$$

OR

$$P_{out} = \frac{1}{2} \left[I_{in} + \frac{1-\epsilon}{1+\epsilon} [Q_{in} \cos(2(\alpha-2\omega)) + U_{in} \sin(2(\alpha-2\omega))] \right] \quad (IAU)$$

1.3 Detector Response in TOAST

Our starting point is a detector quaternion at each sample that rotates the coordinate frame on the sky to the detector frame, with the Z axis along the line of sight and the X axis along the polarization sensitive direction. In the detector frame, the vector along the meridian is offset by an angle of $-\alpha$. Recall that our detector coordinate frame is offset from the overall focalplane frame. The detector frame is also rotated by an angle γ_D . The HWP angle ($\gamma_H(t)$) in the detector frame is time varying and measured from the same reference point:

$$Q_{weight}(45) = \cos(2\alpha)$$

$$Q_{weight}(45) = 0$$

$$U_{weight}(45) = \sin(2\alpha)$$

$$U_{weight}(45) = 1$$

The Stokes Q weight is zero and the Stokes U weight is +1, which makes sense since the detector is aligned with the positive U axis. In the case of a rotating HWP, our Q weight is:

$$Q_{weight} = \cos(2[2(\gamma_D - \gamma_H(t)) - \alpha])$$

(substitute $\gamma_D = \alpha$)

$$Q_{weight} = \cos(2[2\alpha - 2\gamma_H(t) - \alpha])$$

$$Q_{weight} = \cos(2\alpha - 4\gamma_H(t))$$

and the U weight is:

$$U_{weight} = \sin(2[2(\gamma_D - \gamma_H(t)) - \alpha])$$

(substitute $\gamma_D = \alpha$)

$$U_{weight} = \sin(2[2\alpha - 2\gamma_H(t) - \alpha])$$

$$U_{weight} = \sin(2\alpha - 4\gamma_H(t))$$

Consider what happens when the fast axis of the HWP is aligned with the detector orientation ($\gamma_H = \alpha = \gamma_D = 45^\circ$). In that scenario we have:

$$Q_{weight}(45, HWP) = \cos(-2\alpha)$$

$$Q_{weight}(45, HWP) = 0$$

$$U_{weight}(45, HWP) = \sin(-2\alpha)$$

$$U_{weight}(45, HWP) = -1$$

1.3.2 Implementation

Given the previous equations, for both the HWP and non-HWP cases we need to use our detector pointing to determine the α angle at each sample. We can use our detector quaternions to rotate the sky coordinate axes to the detector frame. The resulting direction (detector Z axis) and orientation (detector X axis / polarization sensitive direction) vectors are:

$$\vec{V}_{direction} = \vec{V}_d = Q_{det}(\hat{Z}) = V_{dx} \hat{i} + V_{dy} \hat{j} + V_{dz} \hat{k}$$

$$\vec{V}_{orientation} = \vec{V}_o = Q_{det}(\hat{X}) = V_{ox} \hat{i} + V_{oy} \hat{j} + V_{oz} \hat{k}$$

```
[21]: N = CoordSys3D("N")
      Vdx = sym.Symbol("Vdx")
      Vdy = sym.Symbol("Vdy")
```

```
Vdz = sym.Symbol("Vdz")
Vox = sym.Symbol("Vox")
Voy = sym.Symbol("Voy")
Voz = sym.Symbol("Voz")
```

```
[22]: Vd = Vdx * N.i + Vdy * N.j + Vdz * N.k
Vd
```

```
[22]: (Vdx)îN + (Vdy)ĵN + (Vdz)k̂N
```

```
[23]: Vo = Vox * N.i + Voy * N.j + Voz * N.k
Vo
```

```
[23]: (Vox)îN + (Voy)ĵN + (Voz)k̂N
```

Given these, we can construct the vector orthogonal to the detector line of sight (\vec{V}_d) which is aligned with the local meridian:

$$\vec{V}_{meridian} = \vec{V}_m = V_{dz} \cos\left(\tan^{-1}\left(\frac{V_{dy}}{V_{dx}}\right)\right) \hat{i} + V_{dz} \sin\left(\tan^{-1}\left(\frac{V_{dy}}{V_{dx}}\right)\right) \hat{j} - \sqrt{1 - V_{dz}^2} \hat{k}$$

```
[24]: Vmx = sym.Symbol("Vmx")
Vmy = sym.Symbol("Vmy")
Vmz = sym.Symbol("Vmz")
Vm = Vmx * N.i + Vmy * N.j + Vmz * N.k
Vm
```

```
[24]: (Vmx)îN + (Vmy)ĵN + (Vmz)k̂N
```

This vector is co-planar with the detector X / Y coordinate axes. The rotation angle from the meridian vector to the detector X axis (\vec{V}_o) is the α angle above, and is given by (using the fact that \vec{V}_d is the normal to the plane):

$$\alpha = \tan^{-1}\left(\frac{(\vec{V}_m \times \vec{V}_o) \cdot \vec{V}_d}{\vec{V}_m \cdot \vec{V}_o}\right)$$

Which in terms of components is:

```
[25]: sym.atan2((Vm.cross(Vo)).dot(Vd), Vm.dot(Vo))
```

```
[25]: atan2(Vdx(VmyVoz - VmzVoy) + Vdy(-VmxVoz + VmzVox) + Vdz(VmxVoy - VmyVox), VmxVox + VmyVoy + VmzVoz)
```

Pulling out the minus sign and formatting nicely we get:

$$\alpha = \tan^{-1}\left(\frac{V_{dx}(V_{my}V_{oz} - V_{mz}V_{oy}) - V_{dy}(V_{mx}V_{oz} - V_{mz}V_{ox}) + V_{dz}(V_{mx}V_{oy} - V_{my}V_{ox})}{V_{mx}V_{ox} + V_{my}V_{oy} + V_{mz}V_{oz}}\right)$$

So computing the Stokes weights at each sample involves applying the detector quaternion to two vectors, a few trig functions to get the components of \vec{V}_m , and then the calculation of the α angle.

1.3.3 Stokes Weights Unit Tests

The following plots are generated by the unit tests for different cases of input Stokes parameters and HWP state. For each case in the following plots:

1. The input fake sky has constant $I/Q/U$ values at each pixel, and those are given in the plot subtitle.
2. The plot is “looking out” from the telescope and the COSMO convention Q/U axes are drawn from this perspective.
3. The focalplane frame X axis is parallel to the local coordinate system meridian and pointed “South”. In other words, the detector gamma angle is exactly equal to the “alpha” angle from the local meridian.
4. There are 4 detectors at the boresight, and their actual and expected values are given (which should be the same). I am using a calibration factor of $C = 0.5$, so the total response is $0.5[I + q_{weight} * Q + u_{weight} * U]$. The q/u weights are defined in this document above and in the code for the cases with / without a HWP. The cross polar response is set to zero in this test.

Each row represents a different HWP state (including no HWP at all). Each column represents different input map values of $I = 1, Q = 1, U = 1$, and $I = Q = U = 1$

