Design and Performance Analysis
# of the Poly-Polarimeter

Renia Diamantopoulou

August 29, 2012

MSc Thesis

Contents

1 Introduction

The purpose of the current thesis is to present the design and performance analysis of a versatile ellipsometer called *"The Poly-polarimeter"* . Its primary purpose is for it to be a complete Mueller matrix spectropolarimeter . This extremely versatile instrument will enable us to measure complete Mueller matrices and Stokes vectors with high polarimetric efficiency and high accuracy (4x4 Mueller Matrix with non-diagonal elements $\sim 10^{-3}$ and $\sim 10^{-2}$ diagonal elements)at high speed ($\sim$ seconds).

In astronomical polarimetry it is essential to fully understand the polarimetric properties of the instrument that is used to take the measurements, in order to be able to disentangle real astronomical signals from instrumental effects.

For such astronomical instruments it is highly important to also fully understand the polarimetric properties of the individual optical elements. These instruments introduce two types of polarization effects.One is the instrumental polarisation which is the polarization that is generated by the system and the other effect is the cross talk as one polarization state is transformed into another as the light passes through the instrument.

These instrumental can be caused by reflection and/or refraction processes, birefringent behaviour of material and in general , any deviation of a circularly symmetric optical system (with respect to the optical axis). therefore, it is crucial to characterise the properties of both single optical elements and the complete set up of the optical system designed to perform accurate polarimetry.

"The Poly-polarimeter" will be able to do this operation in the wavelength range of 400-900 nm and has two modes: spectropolarimetry and imaging. The first mode will obtain polarimetric measurements of the sample with a spectral resolution of 2.5nm. The second one will provide polarimetric images of the sample with a spatial resolution of 30 nm (see section 2.1 for detailed description of the scientific requirements). It will also be able to analyse samples in both reflection and trnasmission.

The polarisation properties of the light and the process of measuring it is explained in the current chapter. In chapter 2, the design requirements and the development of *"The Poly-polarimeter"* is discussed. The following chapters describe a comparison between the theoretical predictions, the results from a model and the actual lab measurements. The last chapter of the current thesis presents the conclusions drawn and recommendations for future improvements based on the presented results.

1.1 Introduction to polarisation

1.1.1 Theory of polarisation

Polarisation is a wave property which describes the orientation of its oscillations. It is an important physical property of electromagnetic waves which is connected with the transversality character, with respect to the direction of propagation, of the electric and magnetic field vectors. Under this respect, the phenomenon of polarisation is not restricted to electromagnetic waves, but could in principle be defined for any wave having a transverse character, such as, for instance, transverse elastic waves propagating in a solid, transverse seismic waves, waves in a guitar string, and so on. On the contrary, polarisation phenomena are obviously inexistent for longitudinal waves, such as the usual acoustic waves propagating in a gas or in a liquid.

In general, polarisation is created whenever the symmetry around the propagation direction of the light is broken.This is the case for anisotropic optics such as polarisers and retarders, but this breaking

of symmetry can also be due to the change of direction of the light itself in a scattering process or due to the presence of magnetic fields.[1]

Since we are interested in studying the light as an electromagnetic wave lets consider an electromagnetic, monochromatic plane wave of angular frequency ω. It is propagating in vacuum along a direction that we assume as the z-axis of a right-handed reference system. In a given point of space, the electric and magnetic field vectors of the wave oscillate in the x-y plane according to the following equations: $E_x(t) = E_1 cos(\omega t - \phi_1)$, $E_y(t) = E_2 cos(\omega t - \phi_2)$, where E_1, E_2, ϕ_1, and ϕ_2 are constants. The same oscillation can also be described in terms of complex quantities by writing

$$E_x(t) = Re(\varepsilon_1 e^{-i\omega t}), E_y(t) = Re(\varepsilon_2 e^{-i\omega t}) \tag{1}$$

where ε_1 and ε_2 are given by

$$\varepsilon_1 = E_1 e^{-i\phi_1}, \varepsilon_2 = E_2 e^{-i\phi_2} \tag{2}$$

The polarisation of an electromagnetic wave is defined as the orientation of the electric field vector. Recall that the electric field vector is perpendicular to both the direction of travel and the magnetic field vector. The polarisation is described by the geometric figure traced by the electric field vector upon a stationary plane perpendicular to the direction of propagation, as the wave travels through that plane. An electromagnetic wave is frequently composed of two orthogonal components E_x and E_y as shown in Figure 1.
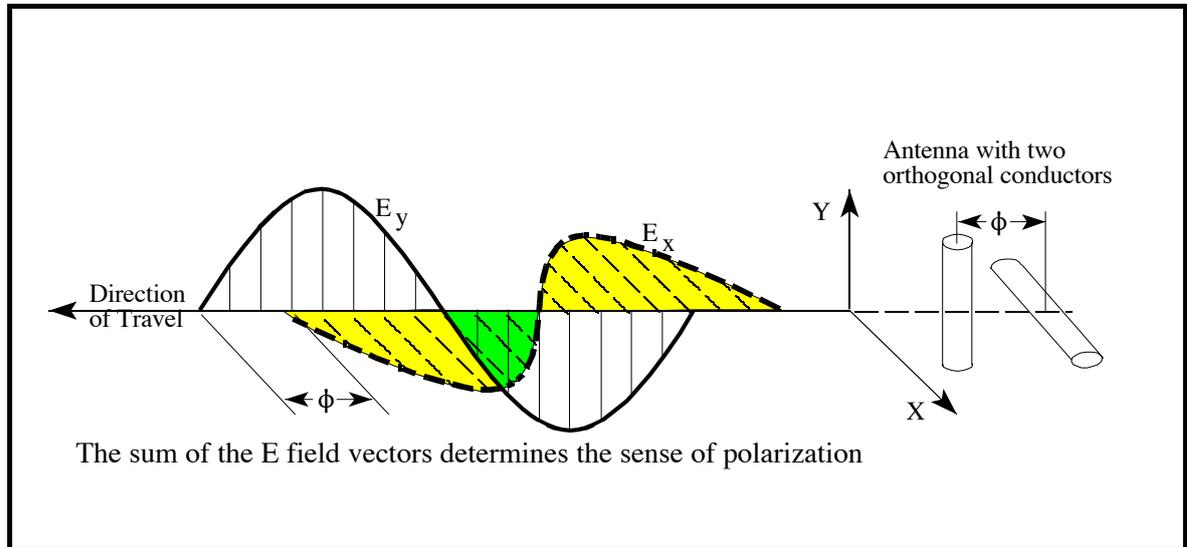
Figure 1: polarisation Coordinates

The geometric figure traced by the sum of the electric field vectors over time is, in general, an ellipse as shown in Figure 1. Under certain conditions the ellipse may collapse into a straight line, in which case the polarisation is called linear. In the other extreme case where the two components are of equal magnitude and 90° out of phase, the ellipse will become circular as shown in Figure 2. Thus

linear and circular polarisation are the two special cases of elliptical polarisation. Linear polarisation may be further classified as being vertical, horizontal, or slant. Figure 2 depicts plots of the E field vector while varying the relative amplitude and phase angle of its component parts. The composition of two orthogonal oscillations of the same frequency gives rise to an ellipse.

As we are mostly interested in studying astronomical objects, we will try to analyse the background theory of processing the obtained data from satellites or telescopes.
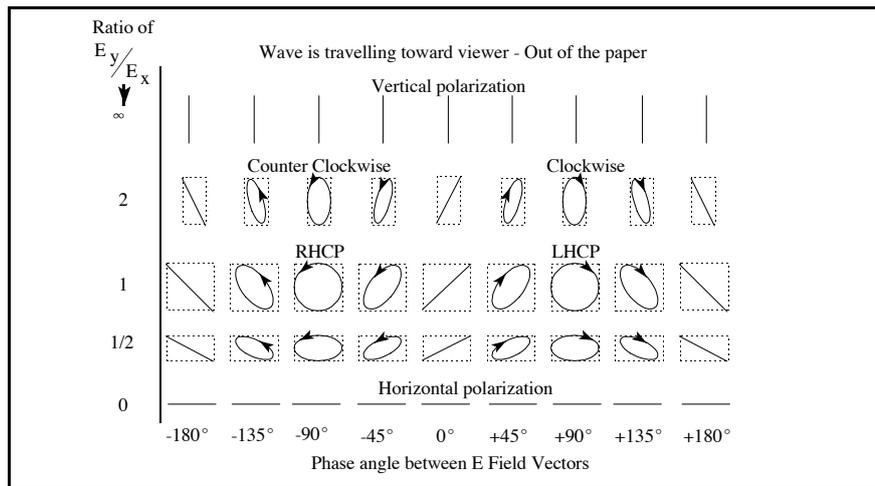
Figure 2: polarisation as a Function of E_y/E_x and Phase angle

1.1.2   Polarimetry/Stokes formalism

All astronomical sources are partially polarised to some degree. In order to characterise the polarisation state from the astronomical source, several formalisms have been developed. Polarisation in astronomical contexts is considered to be best described by the Stokes-Mueller formalism. The polarisation of an astronomical source can be described at first approach with the four-term Stokes vector $S = (I, Q, U, V)$. The Stokes formalism is complete in the sense that it describes all partially polarised polarisation states Figure 3 represents the Stokes parameters. As far as the four parameters in Stokes vector are concerned, I denotes the intensity regardless of polarisation, Q and U describe the two-dimensional state of linear polarisation, and V represents circular polarisation.
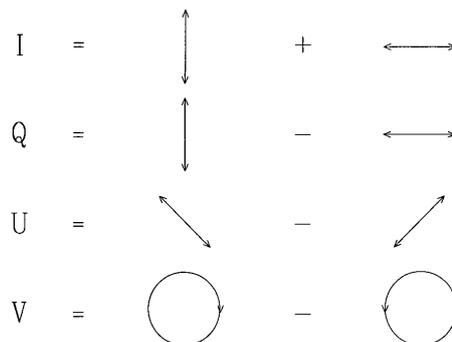
Figure 3: Pictorial representation of the Stokes parameters as seen looking down the beam

The Stokes vector is defined as :

$$\vec{S} = \begin{pmatrix} I \\ Q \\ U \\ V \end{pmatrix} = \begin{pmatrix} \langle E_x^* E_x + E_y^* E_y \rangle \\ \langle E_x^* E_x - E_y^* E_y \rangle \\ \langle E_x^* E_y + E_y^* E_x \rangle \\ \langle i(E_y^* E_x - E_x^* E_y) \rangle \end{pmatrix} = \begin{pmatrix} I_{0°} + I_{90°} = I_{45°} + I_{-45°} = I_{RHC} + I_{LHC} \\ I_{0°} - I_{90°} \\ I_{45°} - I_{-45°} \\ I_{RHC} - I_{LHC} \end{pmatrix}, \tag{3}$$

where E_x and E_y are given by eq. (1), (2).

The linear combination of Stokes parameters for a single intensity measurement after filtering for linear or circular polarisation can be given from the following equations.

$$I = I_{0°} + I_{90°} = I_{45°} + I_{-45°} = I_{RHC} + I_{LHC} \tag{4}$$

$$I = I_{0°} + I_{90°} = \frac{1}{2}(I + Q) + \frac{1}{2}(I - Q) = \frac{1}{2}(I + U) + \frac{1}{2}(I - U) = \frac{1}{2}(I + V) + \frac{1}{2}(I - V) \tag{5}$$

$$Q = I_{0°} - I_{90°} = \frac{1}{2}(I + Q) - \frac{1}{2}(I - Q) \tag{6}$$

$$U = I_{45°} - I_{-45°} = \frac{1}{2}(I + U) - \frac{1}{2}(I - U) \tag{7}$$

$$V = I_{RHC} - I_{LHC} = \frac{1}{2}(I + V) - \frac{1}{2}(I - V) \tag{8}$$

Eq.5-7 are an example of optimal demodulation for the modulation scheme we used in our project (for more details see section 1.2 for definition of modulation/demodulation). Notice the structure of the right term of Eq(5) and Eq(6). It becomes obvious that polarimetry is a differential technique. To verify that the light is linearly polarised in the vertical direction, we should measure the intensity through a vertical polariser but also through a horizontal polariser. The polarised Stokes parameters Q, U, V are all obtained through differences in such measurements. (see i.e. Eq.(6) third part) We have witnessed many cases where the absolute values of $Q, U and V$ are not relevant, therefore the Stokes vector is normalised by the intensity I. With this method the normalised Stokes parameters are independent of the photometric accuracy of the instrument. The most common used forms of the Stokes parameters are the fractional polarisations $Q/I, U/I, V/I$.

By using the normalised values the computation of the degree of polarisation (P) can also be obtained as follows:

$$0 \leq P = \frac{\sqrt{Q^2 + U^2 + V^2}}{I} \leq 1 \tag{9}$$

It would also be wise to define the degree of linear polarisation (P_L) and the angle of linear polarisation (ϕ_L):

$$P_L = \frac{\sqrt{Q^2 + U^2}}{I}; \tag{10}$$

$$\phi_L = \frac{1}{2} arctan \frac{U}{Q} \tag{11}$$

From the definition of Stokes parameters it is easy to conclude that $I^2 - Q^2 - U^2 - V^2 \geq 0$. Note that $I \geq 0$ but Q, U, V can be either positive or negative. In a homogeneous isotropic medium there is no preferable polarisation mode (except of the case of astrophysical magnetised plasmas). However, for a preferred direction there are always two orthogonal modes that can propagate through the medium without changing their polarisation form. Although the polarisation of these two modes remains unchanged, they travel at different velocities in the case of a birefringent medium (the medium has two refractive indices, one for each mode).

In a linearly birefringent medium its modes have linear polarisation. Radiation that has exactly the polarisation of one of those two modes will not be changed, but for any other polarisation angle, or for circular polarisation, the polarisation form will change as the radiation passes through the medium. Circular birefringence on the other hand causes relative phase shifts between two circularly polarised modes.In general, birefringence will be elliptical. [3]

1.1.3 Mueller Matrices

What can be concluded from above is that the Stokes parameters denote the flow of radiant energy in specific vibrations of the electromagnetic field and all four are expressed in the same units. When radiation passes through a medium the state of polarisation may change. The interaction of the Stokes parameters with matter can be described by introducing the 4x4 Mueller matrix with a general structure given by:

$$M = \begin{pmatrix} I \to I & Q \to I & U \to I & V \to I \\ I \to Q & Q \to Q & U \to Q & V \to Q \\ I \to U & Q \to U & U \to U & V \to U \\ I \to V & Q \to V & U \to V & V \to V \end{pmatrix}$$

Thus, the incoming Stokes vector is connected with the incoming vector through the equation[4][5]:

$$S_{out} = M * S_{in}, \tag{12}$$

The first element in every Mueller matrix M[1,1] represents the transmission of unpolarised light. In the cases where we use normalised Stokes parameters then the Mueller Matrix can be normalised with the first element M[1,1] . In a complicated optical system which consists of n elements, each of the n elements is represented by a different Mueller matrix. The total Mueller Matrix of the whole system is given by the equation:

$$S_{out} = M_n M_{n-1} ....M_2 M_1 = M_{total} S_{in}, \tag{13}$$

We have to be extremely careful with the order of the multiplication of the matrices since the elements' 4x4 Mueller Matrices are multiplicative but not commutative. In a 3-D coordinate system formed by [$Q/I, U/I, V/I$], the normalised Stokes vector for fully polarised light is located on a sphere with radius 1.

For almost all conceivable astronomical objects, Mueller matrix is the most general description of the processing of polarised radiation. In case where we are not absolutely certain about all the elements in the Matrices, the safest path to follow is to examine the behaviour of all optical elements contributing to the polarisation and then decide whether we should neglect them or try to measure

them. Some Mueller matrices used in applications express the modification of the Stokes parameters due to the properties of the medium they travel in. They are normally functions of positions within the medium. Another area of application using Mueller matrix is the scattering of sunlight within the planetary atmospheres.

1.2 How to measure a polarised signal

A recent field of growing interest is the detection of exoplanets. Since the signal of exoplanets is polarised we need to figure a really accurate way to calculate the receiving signal. The very first thing one needs to do is to observe the astronomical source (i.e. exoplanets) with a polariser positioned at $0°$ and at $90°$. The intensity measured in each case is different. This process of rotating the polariser to take different intensity measures is called modulation. To complete the process one needs to take at least two measurements and this is also obvious from Eq4-Eq8. In order to receive the signal coming from the astronomical source one needs to practically subtract those two measurement we did with the polariser. This process is called demodulation. So after the modulation and demodulation process what you will result with is the signal from the object you study.

1.2.1 Modulation & Demodulation process

Recalling the definition of the Stokes parameters in equation 4-8 , one requires at least two photon flux measurements to retrieve $I, Q, U or V$. The modulation of the polarimeter describes how the various measurements yield the Stokes vector.

In principle to measure Q and U one could rotate a polariser, however, this makes the measurement to be dependent on the polarisation properties of the optics after and before the analyser. Having a fixed analyser makes the measurement independent of the optics after it, in other words the transmission of the system is reduced. Some times it is necessary to introduce additional polarisation optics acting as modulators. Their job is to convert the polarised signal into the polarisation that analyser actually analyses. Four measurements with a half wave plate @$22.5°, 67.5°$ and @$0°, 45°$ then yield (I±U)/2 and (I±Q)/2). A half wave plate (HWP) which shifts the polarization direction of linearly polarised light (e introduces a phase difference of radians between perpendicular axes) It is of high importance that the read-out of the detector is synchronised to thew modulation sequencing. Such a modulation scheme can be generalised with the use of liquid crystals. [1]

A single measurement of the polarimetric signal is not enough to determine all four Stokes parameters of the incoming radiation. In order to change the polarimetric signal so as to measure all four Stokes parameters one has to change the orientation of the optical axis or insert a rotation to the retarder. Since we are able to measure only intensities, we are restricted to detecting linear combinations of the four Stokes parameters and require several measurements (at least four) for recovery of all four Stokes parameters of the incoming radiation after it passes through an analyser [6].The polarimetric analysis is thus performed modulating the intensity transmitted by the analyser. A simple polariser is shown in figure 4. It can be simply consisted of a linear analyser-polariser and a linear retarder.
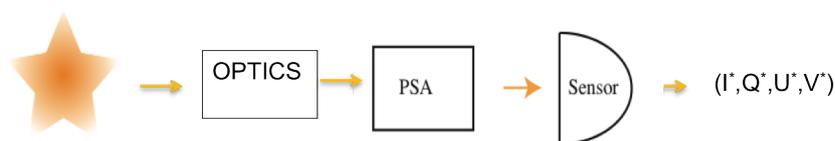

Fig.4. Schematic of a polarimeter.

The general form of a polarimeter to measure the polarisation would look like Fig.4. It consists of a light source, a system of optics (retarders) a polarising state analyser (PSA) which analyses the incoming Stokes vector and a sensor that measures the final outcome polarisation of the system. A PSA consists of two retarders (probably ferroelectric liquid crystals FLCs) and two wave plate in between and a polariser at the end.

In the case of a single detector being used, one can construct a linear system of equations , with the measurements taken at n given times. In general, any sequence of four or more measurements of linear combinations of I, Q, U and V is sufficient, as long as this sequence is non-redundant. A modulation scheme as the one we describe can be best expressed by a n x 4 matrix O :

$$I_{meas} = \mathbf{O}S_{in}, \tag{14}$$

$I_{meas} \equiv (I_{meas}(t_1), I_{meas}(t_2), I_{meas}(t_3), ..., I_{meas}(t_n))^T$ is a polarimetric signal vector contains the n intensity measurements according to each individual modulation state. The matrix $\mathbf{O}$ is the so called *modulation matrix*. The n rows of the modulation matrix are in reality the first rows of the complete Mueller Matrix of the instrument (M_{total}) for each modulation state i. Once crossing this step finding the input Stokes vector involves nothing more than inverting the *modulation matrix*. The measured intensities must be transformed into Stokes parameters using through the demodulation process. Through the demodulation process the measured intensities should be transformed into the Stokes parameters by using the inverse matrix of *modulation matrix* $\mathbf{O}$. In the case of a simple polarimeter when only four measurements are enough to determine all four Stokes parameters the 4x4 *modulation matrix* $\mathbf{O}$ has a unique inverse given by the equation

$$I_{in} = \mathbf{D}I_{meas}, \tag{15}$$

$\mathbf{D}$ is called the *demodulation matrix* of the system.

In the simple case where only four measurements are enough to determine all four Stokes parameters the demodulation matrix is the unique inverse matrix of $\mathbf{O}$ .In other cases (where $n > 4$) an optimum pseudo-inverse demodulation matrix is defined.

$$\mathbf{D} = (\mathbf{O}^T\mathbf{O})^{-1}\mathbf{O}^T, \tag{16}$$

Under ideal conditions the diagonal elements of the matrix product $\mathbf{X = DO}$ are unity.$\mathbf{X}$ represents the errors to the input signal coming from the physical parameters and the demodulation process.

With the demodulation matrix $\mathbf{D}$, one optimises the polarimetric efficiency of the system.

In general , it is easier to design a polarimeter based on liquid crystals (as the one we built in the lab) which is able to generate modulation sequences with optimum efficiencies. For instance, the Stokes definition scheme has optimal efficiencies $\epsilon_{2,3,4} = 1\sqrt{3}$.

The use of this metric of polarimetric efficiency to design modulators that cover a very broad wavelength range was successfully proposed by Tomczyk [7]. The fact that the retardance of wave plates used in modulators is often very chromatic limits the operational wavelength range for such a modulator. By taking the wavelength dependence known for the retardance of liquid crystals for granted we are able to design polychromatic modulators based on liquid crystals with very specific thicknesses and orientations so as to achieve the optimum polarimetric efficiencies over a certain wavelength range. This means that the modulation and demodulation matrices of such modulators

vary strongly with wavelength. Thus, this approach is not suitable for broadband polarimetry but is only applicable for narrow-band or spectroscopic measurements.

At the end of each set up there is a spectrometer translating the output signal. The behaviour of each element used in the set up deviates with respect to the wavelength; some elements are more sensitive to polarisation in some wavelengths. Therefore, a spectrometer efficiently represents the dependence of the system's behaviour on wavelength. After translating the signal through the spectrograph for each modulation state, the modulation and demodulation process of the whole system will begin (having results for all modulation states) predicting the incoming signal.

1.2.2 Polarimetric Efficiencies

Because polarimetric accuracy is a major goal in solar polarimetry , we should look for that modulation scheme that produces minimum uncertainties σ_i for each of the four meausred$I_{in,i}$.

According to del Toro Iniesta to minimise the individual contribution to the uncertainties in the retrieved Stokes parameters and to be able to compare different modulation schemes it is convenient to define the efficiency of the modulation scheme as the four-vector ϵ whose components are

$$\epsilon_i = (n \sum_{j=1}^{n} D_{ij}^2)^{-1/2} \tag{17}$$

Now we can understand that two modulation schemes of equal efficiency will have equal contributions[6].

In our specified case we want to use FLCs for speed. Pancharatnam configurations [5] of FLCs have been attempted but these attempts maybe feasible but not proven successful. These Pancharatnam configurations for achromatic modulators would not be in general full-Stokes and since one of our primary goals is to produce a complete Mueller Matrix modulator we optimise for polychromatic modulator which is full- Stokes in essence.

1.3 Ellipsometry/Mueller Matrix polarimetry

1.3.1 Ellipsometry

Ellipsometry is the measurement of the change in polarisation of a light beam after interaction with a sample. Consequently an ellipsometer consists of a polarisation state generator (PSG), which converts a beam of photons into known Stokes vectors, a sample, and a polarisation state analyser (PSA), also called polarimeter, which converts the polarised light emerging from the sample in intensities in a known way. If an ellipsometer gives a determination of all elements of the Mueller matrix of the sample,(in other words measuring both linear and circular polarisation) it is called a complete Mueller matrix ellipsometer (cMME)

Whereas the general idea of a Mueller polarimeter or in other words an ellipsometer is represented in Fig.8. In this case a polarising state generator (PSG) is added. A PSG consists of the same elements as PSA but placing them in the opposite order (see Fig.5)
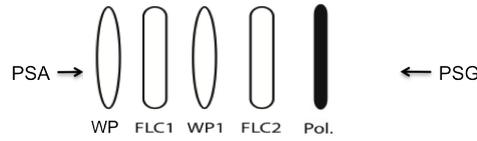
Fig.5. Schematic of the PSA and PSG.

PSG generates different polarization state in each measurement, so that we are aware of the incoming polarisation state. In this case a Mueller matrix will be the output. The Mueller matrix of the sample, of which the first row indicates the transformation of a Stokes vector into intensity.
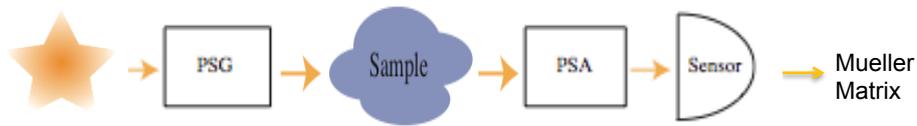
Fig.6. Schematic of the general Mueller polarimeter/ellipsometer.

The are two kinds of complete Mueller Matrix ellipsometer. The one with phase modulation- (PMMME) and one with rotating element- (REMME). Note that a cPSG used in reverse acts like a cPSA and vice versa, so in order to describe cMMEs it is sufficient to focus at cPSGs as shown in Fig.8.

A cPMMME consists of one polariser and two variable phase retarders, all with a fixed orientation, on both sides of the sample. A polariser turns the incoming light into linearly polarised light, with the polarisation direction along the polarisers transmission axis, and the variable retarder changes the phase of the incoming light by retarding the component along the slow axis. The phase change is variable, so it is able to convert various polarisation states into linear polarisation and vice versa. The exact polarisation change depends on the angle between the polariser's transmission axis and the retarder's fast axis. For example, the conversion of linearly to circularly polarised light (retarder is a quarter wave plate) and the mirroring of linearly polarised light around the fast axis of the retarder (half wave plate) are shown in Fig.7. Because of the fixed orientations of the components two retarders are needed, because with only one retarder it is impossible to make both $I + Q$ and $I + U$. [9]
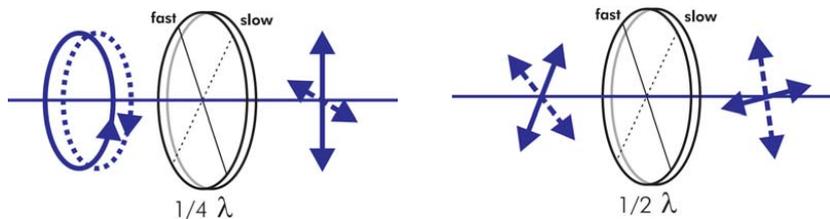
Fig.7. The change of polarization states by a) a quarter-wave and b) a half-wave retarder

A great advantage of the cPMMME design is that there are no rotating components. Measurements can be performed very fast, because it takes about 10 ms for the retarder to change from one retardance position to the other. The disadvantage of variable retarders is the strong sensitivity of the retardance to temperature fluctuations, which causes the need for frequent calibration or temperature control. The PSG of a cREMME contains at least one fixed polarizer and one rotating retarder with a fixed retardance.

2   Design of The Poly-Polarimeter

The first step of the designing is to identify the scientific requirements we would like this instrument to fulfil. The next step is to design a set-up that will cover the principals of the instrument's function. As explained in the introduction to ellipsometry,, the instrument will consist mainly of two parts: aPSG and a PSA. Sine the PSG is a mirrored representation of a PSA we detailed the design for the PSA so we can later apply it for the PSG.

2.1   Scientific Requirements

1. Stokes parameters

    We want to measure all four Stokes parameters (I, Q, U, V)

2. Imaging/spectrometry/combination

    On one side we would want spectrometry so we can analyze the behavior of the system for different wavelength but on the other hand we would like to have imaging ellipsometry so we can characterise single samples. A combination of both would be ideal but we would lose both spectrometry and imaging capability.

3. Wavelength range

    400 nm - 1micron (shorter wavelengths than almost 430nm are difficult to reach as the FLC material will be degraded by exposure to UV light) In the spectrograph version the wavelength range could be 400-800 nm otherwise the images overlap!

4. Spectral resolution

    The general definition of spectral resolution is given by the equation $R = \frac{\lambda}{\Delta\lambda}$. With affordable lab spectrograph $\Delta\lambda$ being usually 1 nm, for a normal mean wavelength $500nm$, $R = \frac{\lambda}{\Delta\lambda}$=500. In imaging, we use narrow-band filters with $\Delta\lambda$ = 10 nm so in this case for a normal mean wavelength $500nm$, $R = \frac{\lambda}{\Delta\lambda}$=50. However, the exact value of the spectral resolution is driven by the spectral features one wants to measure. For example if one wants to measure the resonance feature in aluminum has to take into account that for the wavelength range 100nm-1μm

5. Sensitivity

    Sensitivity actually refers to the noise level and the amount of photons per count or per pixel that should be enough for a signal to be detected. So it would be acceptable a sensitivity of 10^{-4} order which is translated to 10^8photons/pixel. Sensitivity should always be smaller than accuracy.

6. Accuracy

    The accuracy of the measurement depends on the calibration method that we will use for our instrument. Since we want to measure all four Stokes parameters creating an overall Mueller

    Matrix, the accuracy of our measurements is not a single number but a matrix, whose elements depend on the general requirements of the instrument.

$$AC = \begin{pmatrix} 0.01 & 10^{-3} & 10^{-3} & 10^{-3} \\ 10^{-3} & 0.01 & 10^{-3} & 10^{-3} \\ 10^{-3} & 10^{-3} & 0.01 & 10^{-3} \\ 10^{-3} & 10^{-3} & 10^{-3} & 0.01 \end{pmatrix} = \begin{pmatrix} I \to I & Q \to I & U \to I & V \to I \\ I \to Q & Q \to Q & U \to Q & V \to Q \\ I \to U & Q \to U & U \to U & V \to U \\ I \to V & Q \to V & U \to V & V \to V \end{pmatrix}$$

so this means that the diagonal elements $I \rightarrow I, Q \rightarrow Q, U \rightarrow U, V \rightarrow V$ elements of the accuracy matrix is one order higher than the non-diagonal elements which also represent the cross-talk effect. The accuracy depends on the requirements of the system.

7. The ellipsometer should be able to measure samples in both transmission and reflection. The way of doing that is by rotating one arm of the set up around the sample we want to characterise.

8. Imaging Resolution

   The imaging resolution depends on the detector's requirements.Astronomical CCD resolution is commonly measured in FWHM (Full Width Half Max), which is defined as the diameter of a star disk measured at the radius where the photon intensity equals 1/2 of the maximum intensity (located at the centre of the star image).

2.2 Set-up Design

An important feature of the design that we identified very early on is that we like to obtain the spectrum of both polarisation states simultaneously.This will allow us to obtain the total spectrum (by adding them) and also obtain a measurement of the noise (by subtracting them)

The initial idea of a polarisation state analyser is depicted in fig.8
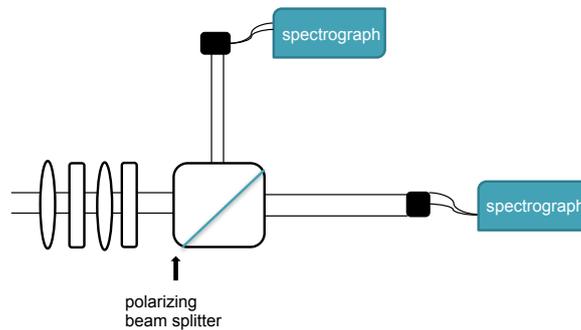
Fig.8. Schematic of the spectropolarimeter.

For the imaging version the set up would look like:
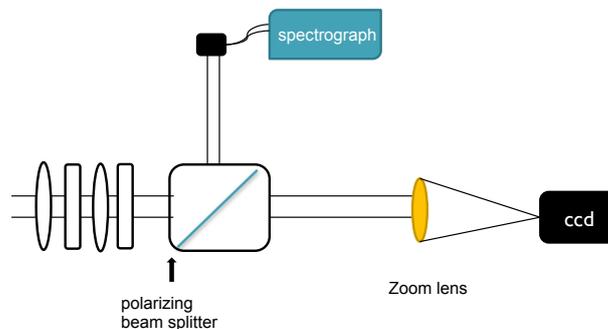
Fig.9. Schematic of our version of an imaging polarimeter.

13

In order to build the ellipsometer we will have to add a PSG to our previous sketches/set ups. Fig.10 shows a spectrograph version of an ellipsometer and Fig.11 an imaging version of it. In horizontal position our instruments will measure samples in transmission. If we , however, flip the left arm of each set up we can easily measure samples in reflection as well.
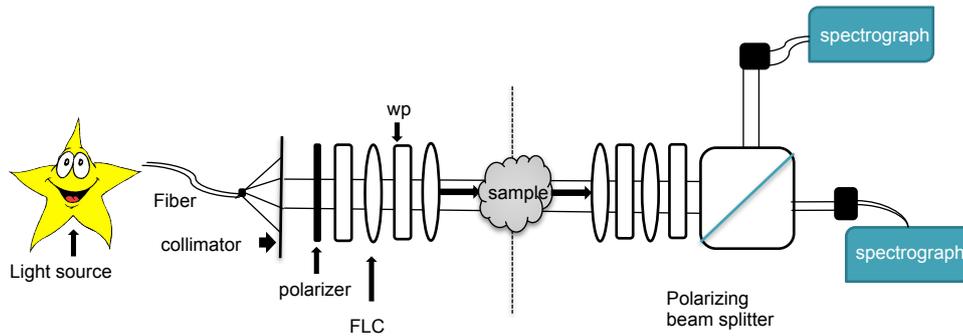
Fig.10. Schematic of the spectrograph version of an ellipsometer.
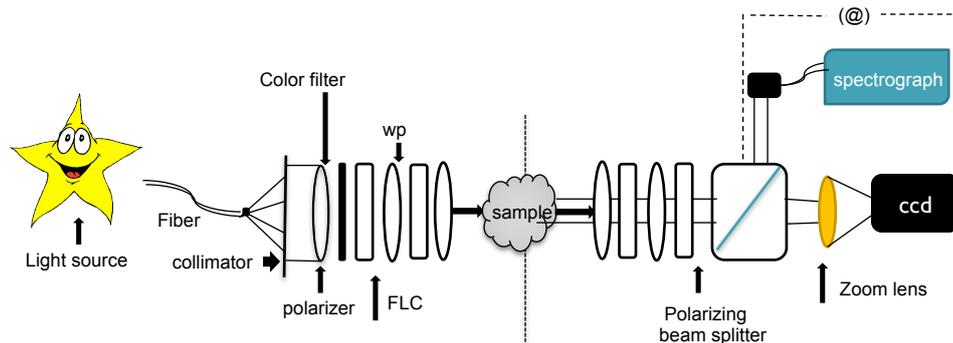
Fig.11. Schematic of the imaging version of an ellipsometer.

The problem with the last version of our instrument is that we are no longer have to deal with a collimated beam because we need somehow to focus the beam before our ccd detector. In this case we are no loner taking measurements of the pupil but the last focus element in our set up which in this case is the fiber. Since we are not interested in fiber but in the sample we have come up with some ideas to solve this problem. This ideas are depicted in Fig.12.a, 12.b and 12.c(for a system).
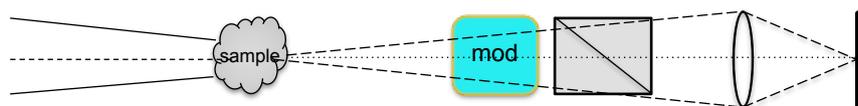
Fig.12.a. Schematic of the first solution in measuring the pupil problem.

In this case we have modified the optics so as to focus the light beam on the sample. The problem is that we no longer have a collimated beam coming into the modulator. The solution to this problem would be to add a focus lens before the modulator so as to collimate the light beam right after the

sample and before the modulator (see Fig.12b). This can work perfectly well not only for a single sample but for a whole optical system (see Fig.12c).
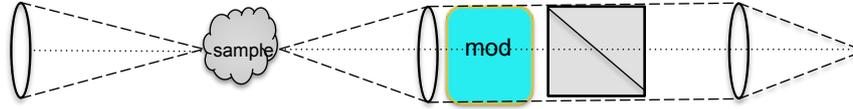
Fig.12.b. Schematic of the second solution in measuring the pupil problem.
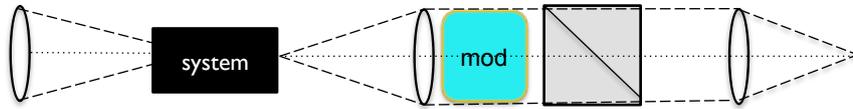
Fig.12.c. Schematic of the second solution(for a system) in measuring the pupil problem.

2.3   Calculation of Photons for Normalised Stokes parameters

It is already introduced in the first chapter the Stokes vector deriving from our system, Eq.3. This is a differential technique so we need to estimate the number of photons needed to have a reliable signal to noise ratio. So based on the accuracy matrix of the normalised Stokes parameters $Q/I, U/I, V/I$ we can calculate the number of photons necessary for the measurements.

$$AC = \begin{pmatrix} 0.01 & 10^{-3} & 10^{-3} & 10^{-3} \\ 10^{-3} & 0.01 & 10^{-3} & 10^{-3} \\ 10^{-3} & 10^{-3} & 0.01 & 10^{-3} \\ 10^{-3} & 10^{-3} & 10^{-3} & 0.01 \end{pmatrix} = \begin{pmatrix} I \to I & Q \to I & U \to I & V \to I \\ I \to Q & Q \to Q & U \to Q & V \to Q \\ I \to U & Q \to U & U \to U & V \to U \\ I \to V & Q \to V & U \to V & V \to V \end{pmatrix}$$

$$\frac{Q}{I} = \frac{\frac{1}{2}(I+Q) - \frac{1}{2}(I-Q)}{\frac{1}{2}(I+Q) + \frac{1}{2}(I-Q)} \tag{18}$$

If we assume that we have no serious polarization ,then $Q \ll I$

$$\frac{1}{2}(I+Q) = a \simeq N, \frac{1}{2}(I-Q) = b \simeq N \tag{19}$$

$$\frac{Q}{I} = \frac{a-b}{a+b} = Y \tag{20}$$

So

$$\delta Y = 10^{-4} = \sqrt{(\frac{\vartheta Y}{\vartheta a})^2 a + (\frac{\vartheta Y}{\vartheta b})^2 b} = \sqrt{\frac{4b^2}{(a+b)^4}a + \frac{4a^2}{(a+b)^4}b} \tag{21}$$

if we substitute in eq.24 a and b with N then we get:

$$\delta Y = 10^{-4} = \sqrt{\frac{1}{2N}} \tag{22}$$

So in other words the number of photons needed per parameter or per modulation state is $N = 10^8/2$. For spectrograph we need $4xN = 4x(10^8/2) = 2x10^8$. For imaging we need $16xN = 16x(10^8/2) = 8x10^8$. The reason for that is that we need 4xN for all Stoked parameters (full Stokes vector) and for a Mueller Matrix we need 16xN.

(For more Detailed calculation of Photons see Appendix A)

2.4　Modulation

In the scientific requirements we already mentioned that we want to measure all Stokes parameters in a wide wavelength range. For that reason we have to introduce the polychromatic modulation in order to achieve the optimal polarimetric efficiencies for all wavelength range.

The process of Polychromatic Modulation should be introduced to analyse the behaviour of the incoming astronomical signals. There are three reasons why the Polychromatic Modulation is chosen in our case: 1) The wavelength range in which we work is too large to achromatise the FLCs 2) Our goal is to measure full Stokes.

2.4.1　Polychromatic Modulation

Since we are about to study a wide λ range we would like our polarisation modulation scheme to be efficient through the whole λ range. We can achieve the best efficiency by using a super-achromatic modulator which properties are independent of λ but it can be sufficiently efficient at one λ. In order to accomplish an acceptable efficiency at all wavelength range we can use polychromatic modulators in such a way to achieve the optimal efficiency at all wavelengths. The modulation matrix of these polychromatic modulators can be totally arbitrary and strongly λ dependent. A solution to optimise the efficiency of their polarimeter over a broad wavelength range is by adjusting the orientation of the fast axes of the retarding elements.The chromatic nature of FLCs creates the deviation in efficiency at different wavelengths. It can,however, be minimised during the wavelength calibration procedure.

The susceptibility to noise of a polarimeter is described by the polarimetric efficiency. For a narrow wavelength range achromatic modulation would be the obvious solution. However, for a wide wavelength range we can also achieve an optimal efficiency using polychromatic modulation. (S. Tomczyk et al. 2010). The maximum possible efficiencies for a balanced modulator with equally important Stoles parameters are: $\epsilon_Q = \epsilon_U = \epsilon_V = 1/3 = 57.7\%, \epsilon_I = 1$.

2.4.2　Modulation Scheme for FLCs

At first we place all four FLCs in the initial mode/position. This means that the first one is on vertical position, the second one is tilted by 45 °,the third and the fourth one are in vertical position. The first two refer to the PSG (polarising state generator) and the last two FLCs correspond to the PSA(polarising state analyser). The general modulation scheme for all possible combinations of FLCs are depicted in Fig.13.

16

FLCs

Black: vertical position   White: 45º tilted

(t)
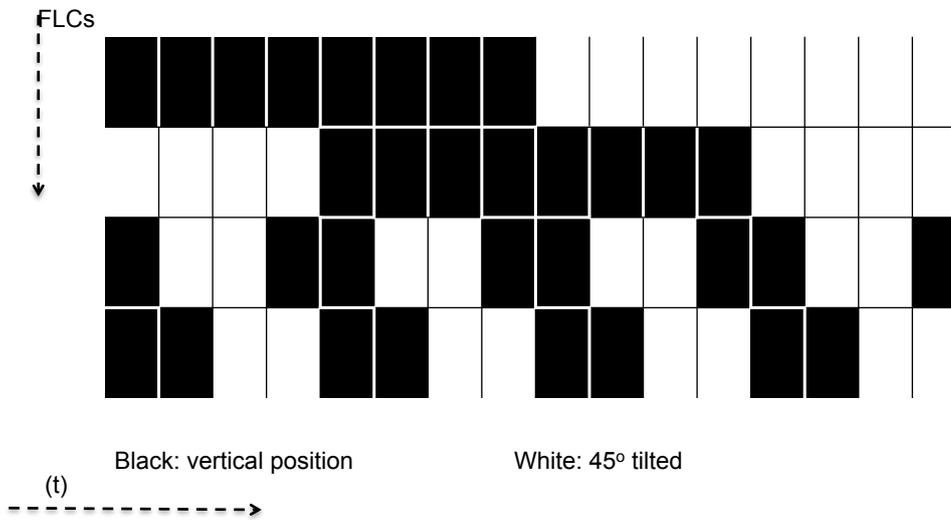
Fig.13 Modulation scheme of the FLCs

The spectral efficiency of our polarimeter is numerically optimised by tuning the retardances and orientations of the optical components: 2 FLCs with 45 cone angle and 2 quarter wave plates. (see Fig.14).
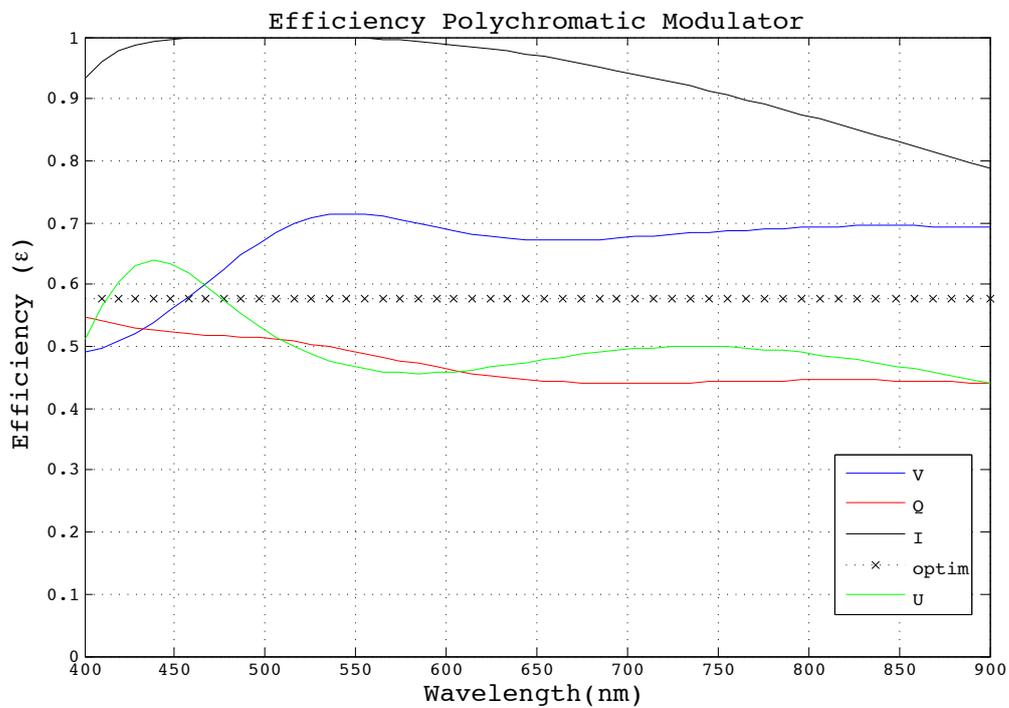
Fig.14 Result of the numerical algorithm to optimise
the efficiencies of the optical components.

The specific design of our modulator regarding the individual elements we used is shown explicitly in Poly-polarimeter set up in APPENDIX B. The theoretical prediction of the Modulation Matrix resulting from this configuration for the PSG is represented in the figures below.

$I \to I$

$I \to Q$

$I \to U$

$I \to V$

$Q \to I$

$Q \to Q$

$Q \to U$

$Q \to V$

$U \rightarrow I$

$U \rightarrow Q$

$U \rightarrow U$

$U \rightarrow V$

$V \rightarrow I$

$V \rightarrow Q$

$V \rightarrow U$

$V \rightarrow V$
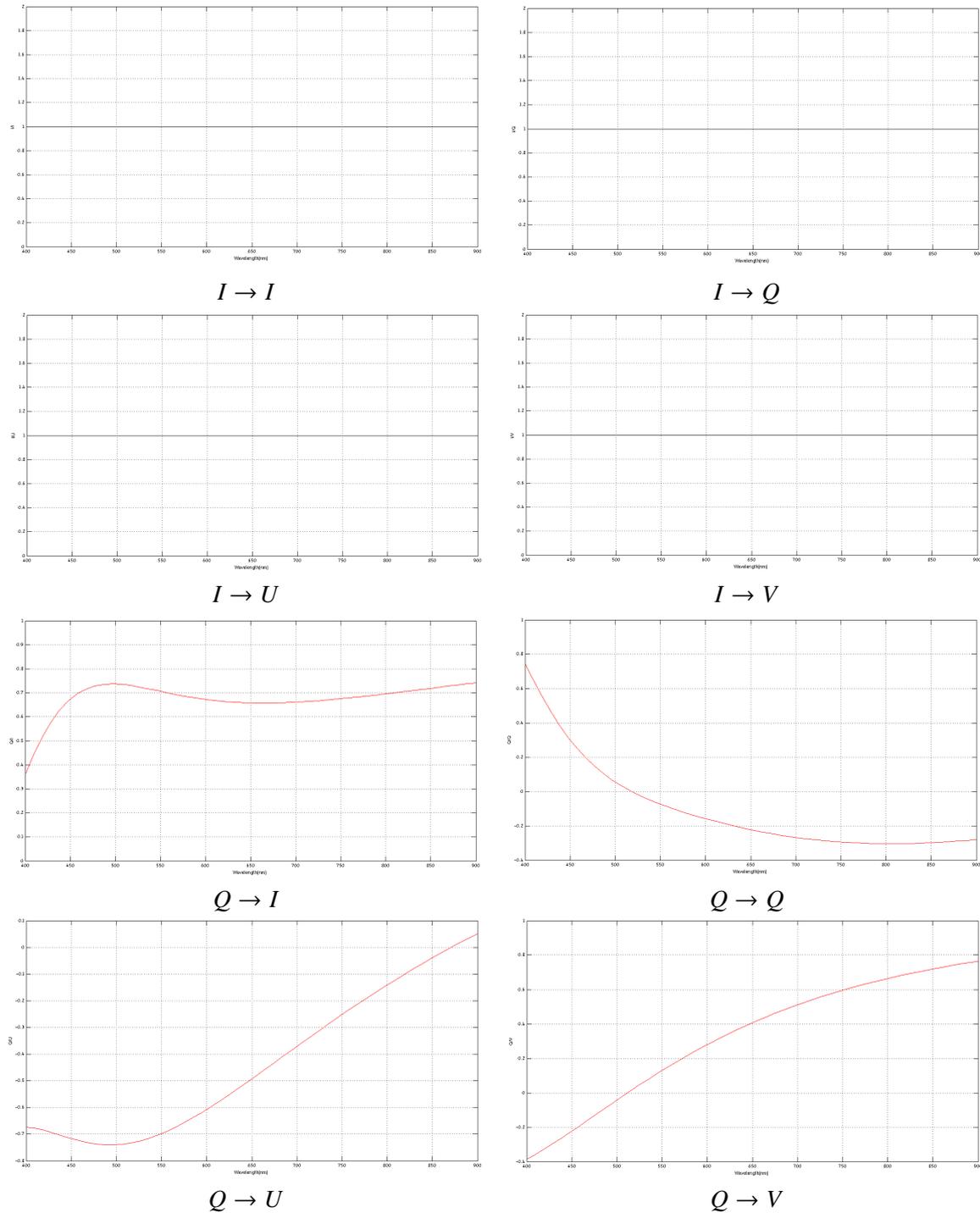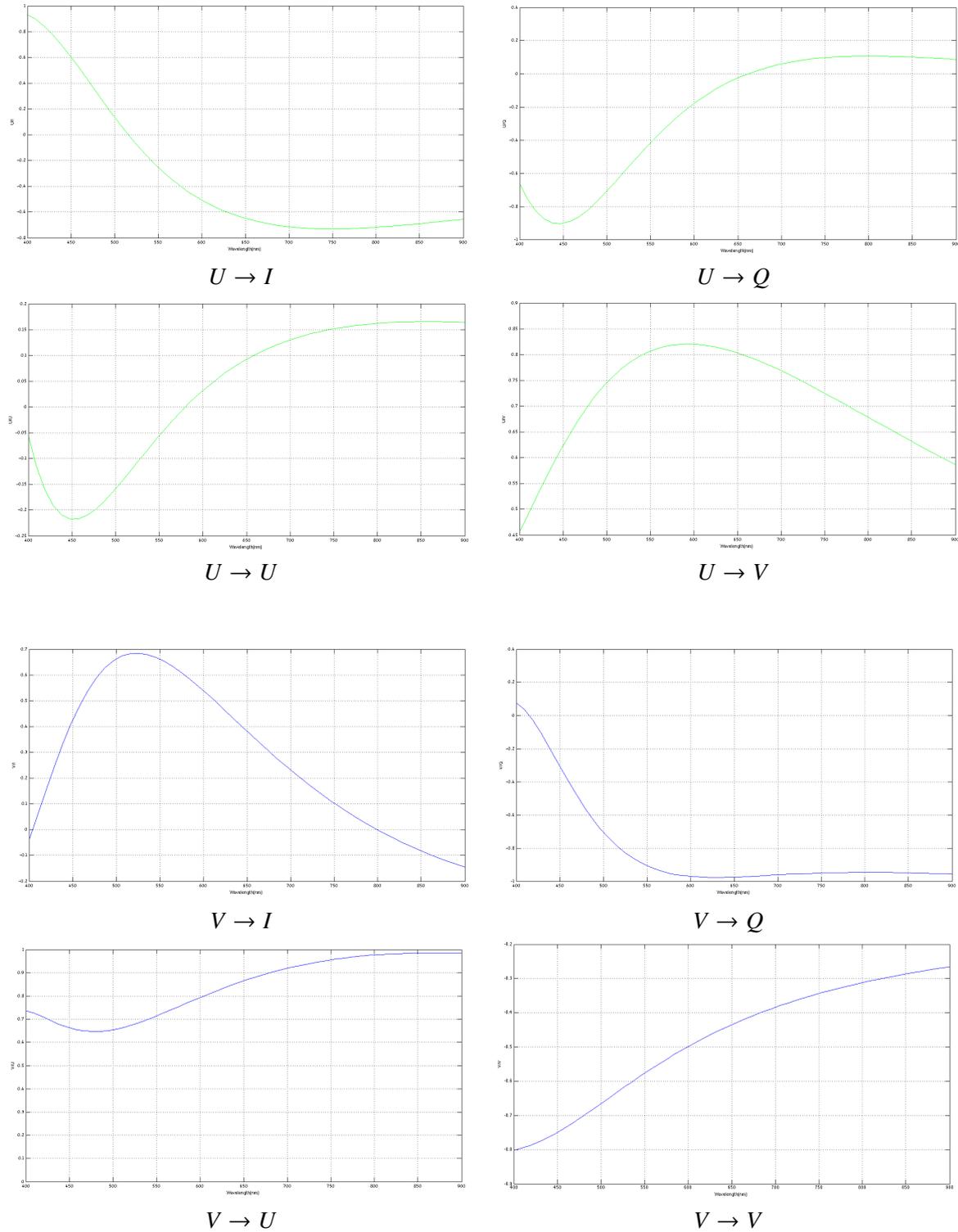
Fig.15 Modulation Matrix Elements with wavelength

2.5 Model Development(M&m's Code)

For modelling the polarimetric behaviour of our instrument we used the M&m's code (de Juan Ovelar et al.,2011). The M&m's code is an error budgeting tool to characterize the behaviour and calculate the accuracy of a system.This very generic polarimetric tool gets as input the MM of each individual element of the instrument and calculates the total MM of the system, the modulation and demodulation matrix coming out of the system plus the errors coming from the dependence of the optical elements on the physical parameters (i.e. temperature).

2.5.1 Poly-Polarimeter's Modulator code

The M&m's code includes all the potential error contributions, although in our case we are not interested in the error behaviour but only in the total Mueller Matrix , the modulation **O** and demodulation **D** Matrix of our system (two wave plates and two FLCs). The modified code used for our system is in details in Appendix C along with the internal functions representing the behaviour of a polariser (RPPOL), a compound wave-plate (compWP) and a wave-plate(WPb) as long as the rotation (ROT) for the polarisation sates to be in accordance after passing through the optical elements. For further information about the Mnm's code you could see reference [11]. So to summarise the most important elements of the code I present here in a compact way all the individual parts of the elements foe building the modulator's code. As we have already mentioned above our modulator consists of two QWP and 2FLCs in between.

1. QWP $compWP(rot_{ang}, alignang, thick1, thick2, wave, name)$

   $compWP(161.102, 90, 421.1, 337.8, wave, "name")$

   It is represented by the inner function $compWP$ which takes as inputs the rotation angle of the WP the alignment angle with respect to the beam, the thickness of the two layers of the material, the wavelength range in which we work and a name to specify the WP.

2. FLC

   $\rightarrow$ rotation before the FLC $ROT(thetadeg)$

   It is represented by the inner function ROT which takes as input the rotation angle of the FLC for the two states.

   $\rightarrow$ Wave plate $WPB(b, dwp_{mic}, wave, wpmat)$

   It is represented by the inner function WPB which takes as input the birefringence of the material, the thickness, the wavelength range and the wave plate material. This shows the retardance in the FLC.

   Because the FLC introduces instrumental polarisation (see chapter 3: Characterisation of FLCs), its behaviour is complicated due to the polarised fringes in other words the anomaly of the reflections within the FLC device. So it actually consists of two polarisers.

   $\rightarrow$ FLC polariser $rppol(rotang, k1, k2)$ Calculates the Mueller Matrix for a Rotating Partial Linear Polarizer represented by the inner function $rppol$

   given the following parametersIt is :

   @param [in] rotang is angle of optical axis relative to the +Q axis of the system
   @param [in] k1 is ratio of transmitted light (Iout/Iin) in the direction parallel to the optical axis

@param [in] k2 is ratio of transmitted light (Iout/Iin) in the direction perpendicular to the optical axis

@ returns P is rotating partial polarizer mueller matrix

The definition of the matrix can be found in Introduction to Spectropolarimetry, J.C. del Toro Iniesta, Cambridge University Press, 2003

$\rightarrow$ FLC polariser $rppol(rotang, k1, k2)$

$\rightarrow$ rotation after the FLC $ROT(thetadeg)$

3. QWP $compWP(rot_{ang}, alignang, thick1, thick2, wave, name)$

   $compWP(106.543, 90, 421.1, 337.8, wave, "name")$

4. FLC (same as 2.)

5. Polariser/Analyser $Analyzer = RPPOL(0., 1., 0.)$

   In this case a single polariser is used at the end of the set up to analyse the incoming polarisation.

   The result for the Modulation Matrix **O** is shown in Fig.16
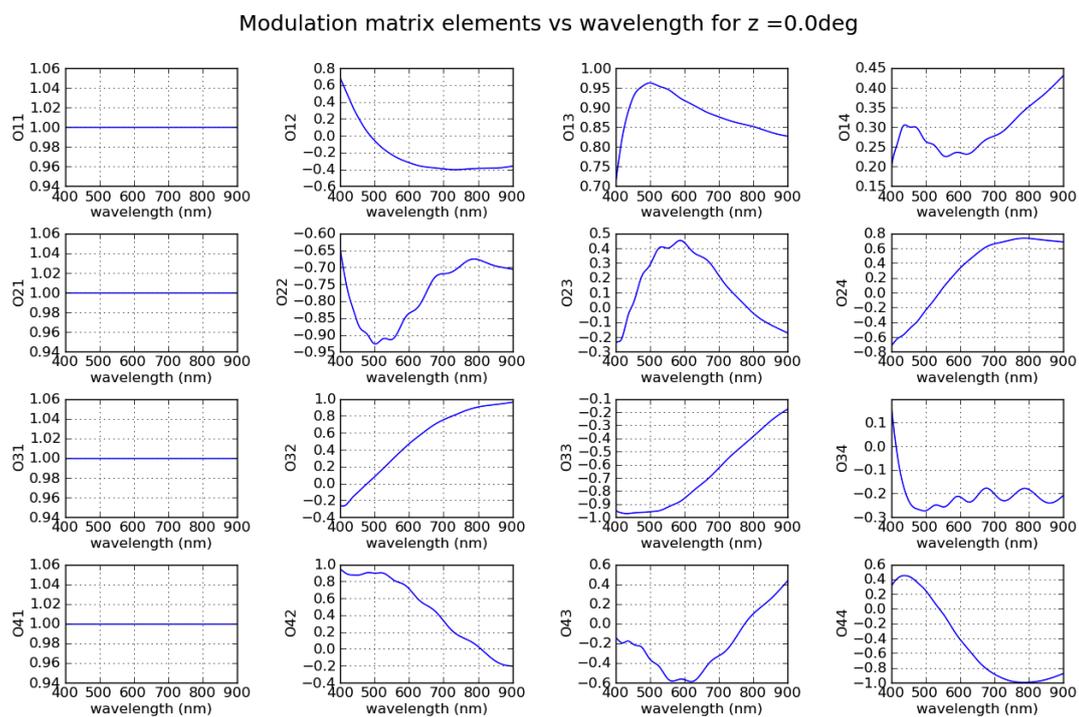
Fig.16 Model's Modulation Matrix Elements with wavelength

3 Characterisation of FLCs (Ferroelectric Liquid Crystals)

Sensitive polarimeters require retarders whose retardance can be quickly varied. This can be achieved either by changing the geometrical thickness; necessarily associated with mechanical motion; of the optical elements or by changing the birefringence (i.e. liquid crystals). Liquid crystals' smetic phases are characterised by well-defined layers that can slide over one-another. The molecules are positionally ordered in one direction. In a smetic phase called C the molecules are tilted normally away from the preferred direction. The ferroelectric liquid crystals (FLC) are the tilted phases of the molecules having a permanent polarisation, the reason why the are called ferroelectric. They respond much more quickly to externally applied fields and can be used, as in our case, to build fast, bistable electro-optic devices when place between closely-spaced electrically conducted glass plates.

The Ferroelectric Liquid Crystals (FLCs) are the most suitable candidates for our instrument. First of all the are really fast, they can provide a polychromatic modulation efficient for every wavelength when combining them with a waveplate in between. The most significant reason, nevertheless, is that the only parameter sensitive to temperature is the switching angle. This facilitates our calibration procedure somehow and since we are already facing a complicated system the calibration could be rather time consuming and inaccurate.

3.1 General behaviour of FLCs

FLC variable retarders act like retarders with a fixed retardation where the direction of the fast axis can be switched by about 45 ° (switching angle) by alternating the sign of the applied external electrical field. They are really fast ($\sim 150 \mu s$) true zero order retarders.
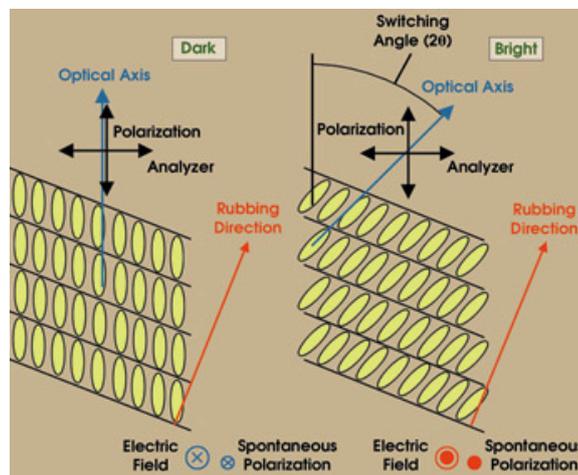
Fig.17 Modulation scheme of the FLCs

3.2   Retardance Characterisation

Generally the FLCs are built of a thin layer of a liquid crystal is sandwiched between two glass plates. By selecting the appropriate thickens d, the layer functions as a half-wave plate (retardance $= \delta = \pi$ or 0.5 in waves) at a distinct wavelength. The optical axis of the liquid crystal layer has two preferred orientations, which differ from each other by approximately 45°. The orientation of the optical axis can be controlled by an external applied electrical field across the FLC layer. Thus, one of the two states can be selected by applying the appropriate voltage.

The value of the switching angle θ mainly depends on the temperature and varies from several degrees to $\theta \approx 90°$ in some FLCs. The temperature dependence usually observed is

$$\theta(T) \propto (T_c - T)^{1/2} \tag{23}$$

whete T_c is a constant. This thermal behaviour has been probed to find the temperature of the required 45° (for the FLCs) and to get an idea of how precisely the FLC temperature has to be stabilised to ensure a proper polarisation modulation. This temperature dependence law is shown in Fg.18. This is the figure we used to determine from the slope of the curve the retardance of a FLC in this temperature scale (room temperature).[12]
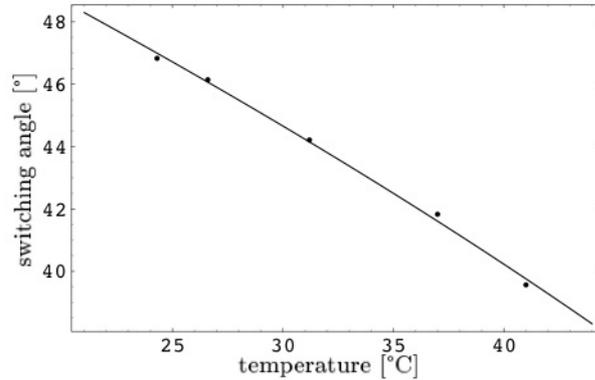
Fig.18 Temperature Dependence of the incidence angle θ of the FLCs

FLCs are only birefringent at optical wavelengths. The dispersion law for the optical birefringence , Δn, follows the Cauchy formula

$$\Delta n(\lambda) = \Delta n(\infty) + \frac{C_1}{\lambda_0^2} + \frac{C_2}{\lambda_0^4} + \frac{C_3}{\lambda_0^6} + ... \tag{24}$$

Δn , C_i are free parameters [12]. Once one becomes familiar with the optical birefringence of the FLC can easily compute the appropriate thickness using the equation for retardance:

$$\delta = \frac{2\pi \Delta n d}{\lambda_0} \tag{25}$$

The desired flexibility of the polarimeter determines the selection of ferroelectric liquid crystals as polarisation modulators. The tuneable birefringence of these elements allows specific wavelength calibration, rotation of the axes of linear polarisation and compensation of instrumental cross-talk. The emergent intensity of unpolarized light with wavelength λ through a configuration with the retarder of equation:

$$\delta = \frac{\Delta nd}{\lambda} \tag{26}$$

,where $\Delta n = (n_e - n_o)$ and δ represents the induced phase retardance expressed in waves.The Mueller matrix of a retarder with its slow axis at 45o and parallel to the direction of negative Q is respectively given by:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\delta & 0 & -sin\delta \\ 0 & 0 & 1 & 0 \\ 0 & sin\delta & 0 & cos\delta \end{pmatrix}$$

and

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & cos\delta & -sin\delta \\ 0 & 0 & sin\delta & cos\delta \end{pmatrix}$$

The emergent intensity of unpolarised light with wavelength λ through a configuration with the retarder of Eq.29 between two parallel and crossed polarisers respectively is given by:

$$I_{\parallel} = I_0 cos^2[2\pi\frac{d\Delta n}{\lambda}] = \frac{1}{4}I_0(1 + cos\delta) \tag{27}$$

$$I_{\perp} = I_0 sin^2[2\pi\frac{d\Delta n}{\lambda}] = \frac{1}{4}I_0(1 - cos\delta) \tag{28}$$

In order to characterise all FLCs, they were placed one at a time between parallel and crossed polarisers with its axes at 45°. By using eq.30 and eq.31 the retardance of a FLC can be given by the equation:

$$\delta = arccos\frac{(I_{\parallel} - I_{\perp})}{(I_{\parallel} + I_{\perp})} \tag{29}$$

That was the exact method we used to define the retardances for all 4 FLCs we used in our experiment. The result from this method is depicted in the following figures.The strange effect on Fig.22 below 400 nm is due to the strange behaviour of the arccos function when approaching really low values. Of course in each calculation for the retardance we subtract the Dark measurement. Since the retardation of the FLCs depends on the thickness of the liquid crystal layer(i.e.see eq.30), the wavelength where the FLC is exactly a half-wave plate (design wavelength) is slightly different for each FLC. In order to find the specific wavelength where the FLCs of our set-up behave as a half wave plate we are using Fig.19. The design wavelength is reached when the efficiencies of Q and U are maximised and the cross-talks disappear (or minimise). In fact the design wavelength can be defined with an accuracy of 3 nm as the FLCs do not behave as pure retarders.

FLC#1   *FLC*#2

FLC#3   *FLC*#4
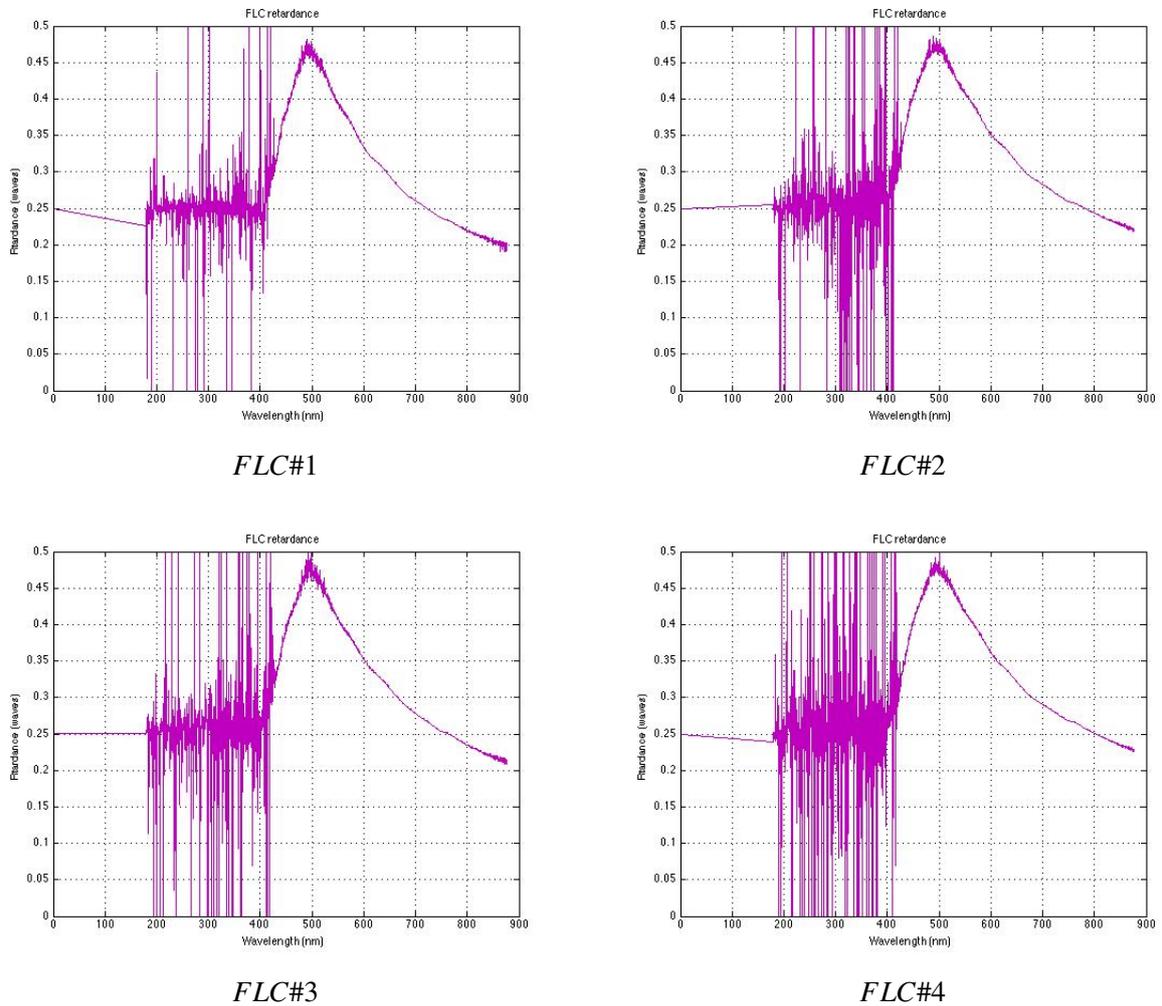
Fig.19 Retardance plots of the four FLCs of the "Poly-Polarimeter" set-up

The Fig.20 that follows depicts the optimal retardance plot calculated based on Fig.19. We know that the FLCs are half wave plates so the retardance should be equal to 0.5(in waves) at some central wavelength. In order to find this wavelength we had to prolong the retardance in order for it to reach the expected 0.5 value at some wavelength.
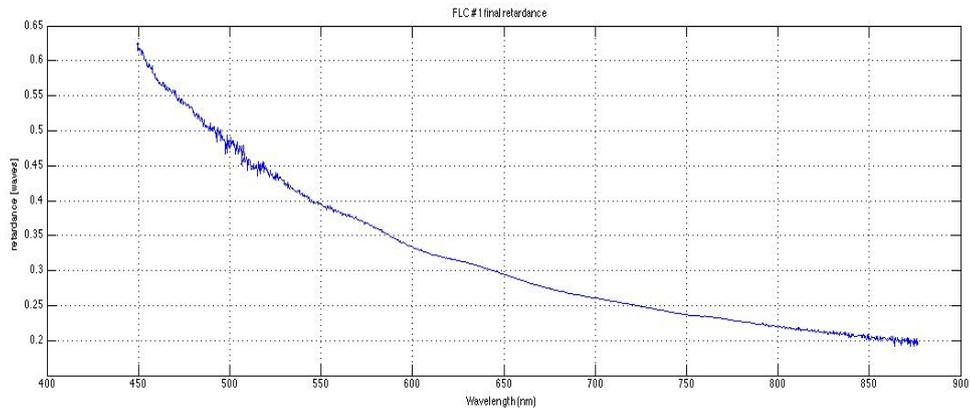
final retardance plot for FLC#1
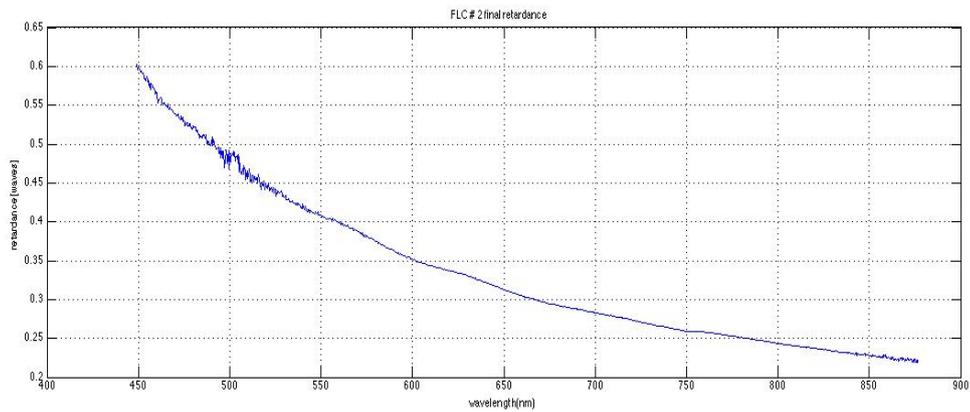
final retardance plot for FLC#2

Fig.20a "Ideal" retardance plots of the four FLCs of the "Poly-Polarimeter" set-up
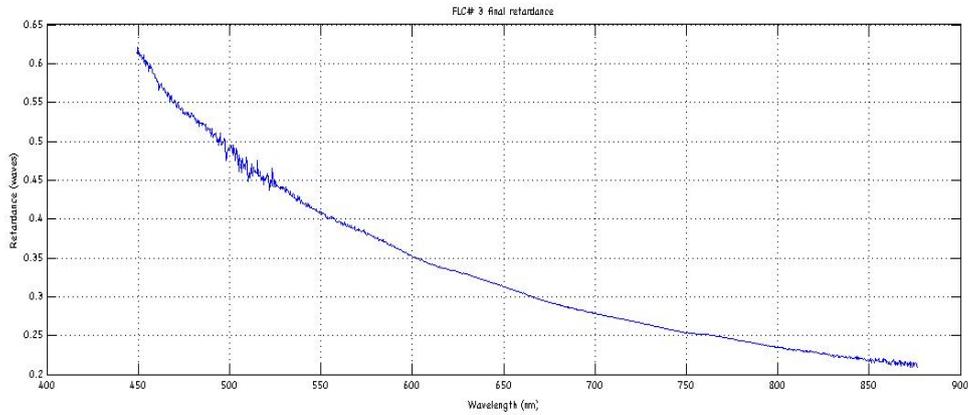
final retardance plot for FLC#3
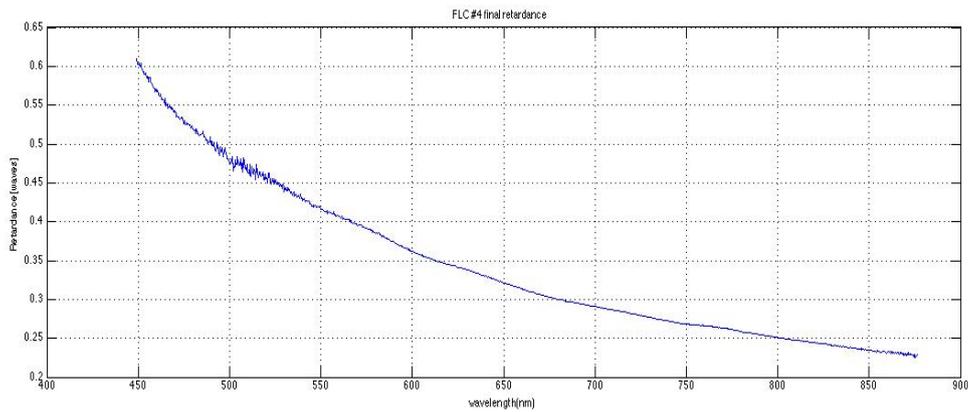
final retardance plot for FLC#4

Fig.20b "Ideal" retardance plots of the four FLCs of the "Poly-Polarimeter" set-up

3.3   Polarised Fringes

One important anomaly are reflections within the FLC device. They induce some partial polarisation which slightly shifts the maxima and the zero-crossing of the matrix elements.[12] Multiple reflections within phase plates often lead to spectral variations of intensity. The study of the spectral variation of the FLC transmittance permits to get an idea of the magnitude of internal reflections within the device. In order to determine the polarised fringes the FLCs introduce to the incoming light, the FLC was illuminated with unpolarised light and two consecutive wavelength scans were made:one with the fast axis and one with the slow axis (FLC @ 0°and 45°).

From Eq.25 $(I+Q)$ represents state1 of FLC and $(I-Q)$ represents state 2. So in order to calculate the polarised fringes caused by the FLCs (Q/I) we have in no other words to add and subtract the measurements of the two different states of the FLCs. The results from the polarised fringes of the four FLCs we used are summarised in Fig.24a and Fig.24b. In order to understand better the outcome we tried to compare the result with the model. (This comparison is shown in fig.21b for FLC#2)
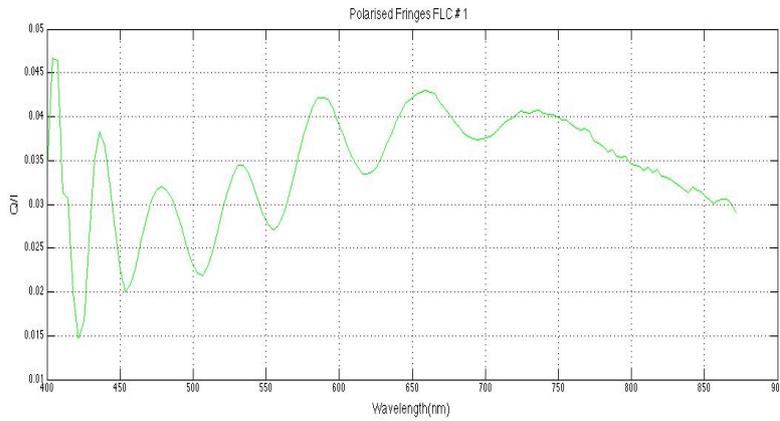
*Fig.21a polarised fringes for FLC#1 of
the Poly − Polarimeterandcomparisonwiththemodel*
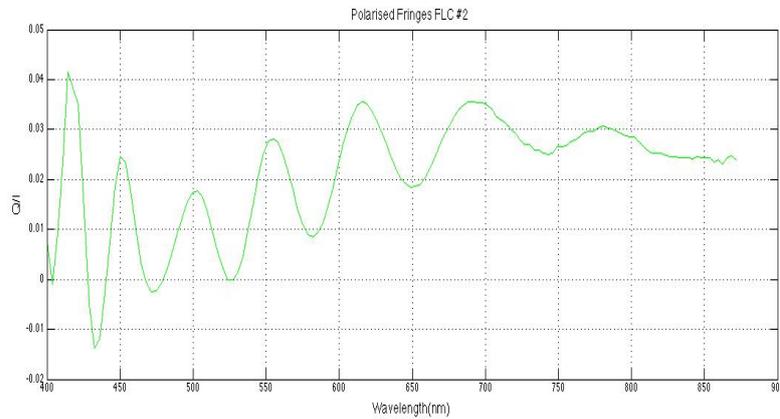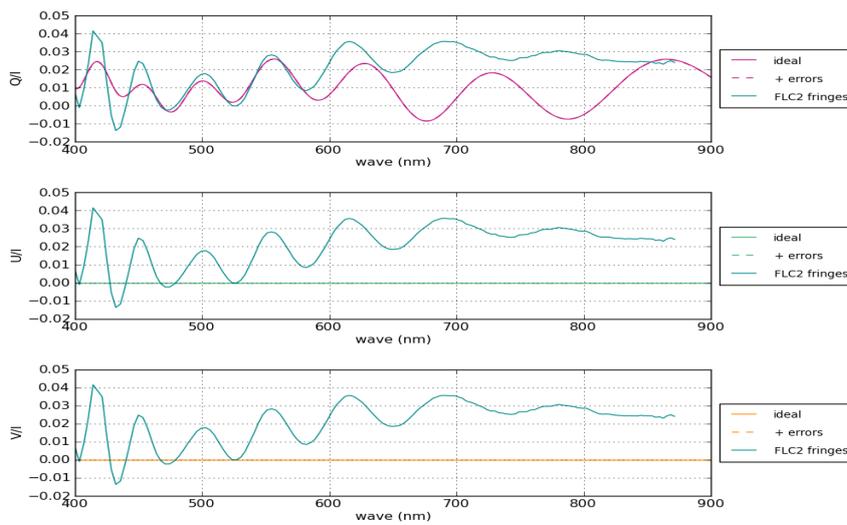
*Fig.21b polarised fringes for FLC#2 of
the Poly − Polarimeterandcomparisonwiththemodel.*
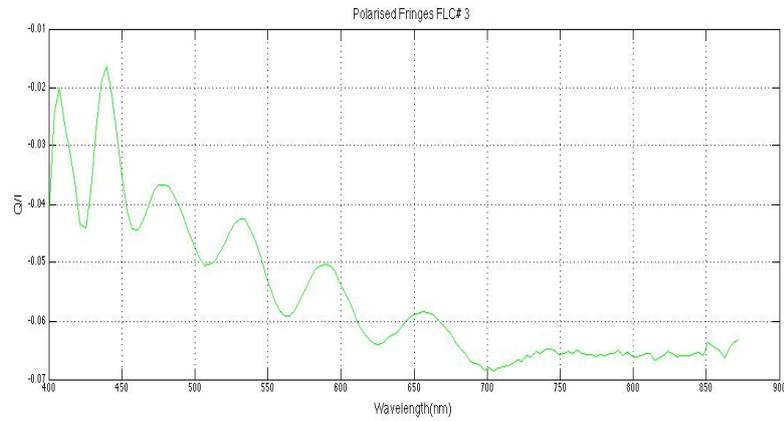
*Fig.*21*c polarised fringes for FLC#3 of
the Poly − Polarimeter*
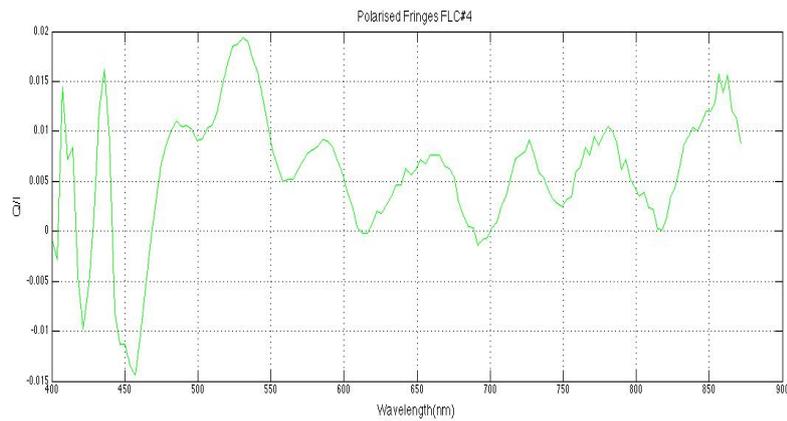
*Fig.*21*d polarised fringes for FLC#4 of
the Poly − Polarimeter*

4   Experimental Results

4.1   Experimental Process

Before I explain the measurement process I will present a specific description of the instrument and the positions of the individual elements of the modulator we built at the lab(see Fig.22). We based the final design of the instrument on the requirements we set before and the results of the numerical algorithm to achieve the optimal efficiency with the best possible combination of the elements we bought.

According to the design of the poly-polarimeter set-up (see Appendix B) in order to achieve the best polarimetric efficiency we have to use FLC#4 and FLC #1 for the polarisation state generator and FLC#3 and FLC#2 for the Polarisation state analyser. After determine the fast axis of each element we built the modulator as shown in Fig.22 using 2 Hale QWP one @ 670 and one @488 nm.

In the end we built the PSG (polarisation state generator) using the FLC # 4 and FLC #1 with the angles shown in APPENDIX B after determined the axis of the FLCs. In order to find the axis of the elements and specifically for the FLCs, we inserted each element between two crossed polarisers and by the output spectrum we determined the position at which the intensity reached the minimum value.
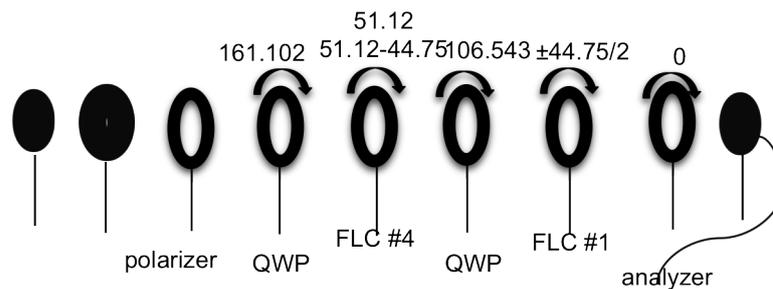
Fig.22 Poly-Polarimeter's Scheme

With the the fast axis of both wave-plates being vertical to the direction of the incoming light beam we place the first polariser of the set up as the light come in at 0° to take measurement for Q and at 90° to measure $-Q$. We place the first polariser of the set up as the light come in at 45° to take measurement for U and at −45° to measure $-U$.

For each of this measurements we detected 4 different spectra corresponding to the 4 different modulation states of the 2 FLCs (see fig.23 which is actually part of the fig.13 of the 4 FLCs)
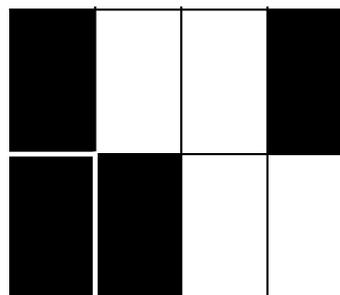
Fig.23 FLC modulation states

An example of Q measurement for detecting the 4 spectra is shown in Fig.24
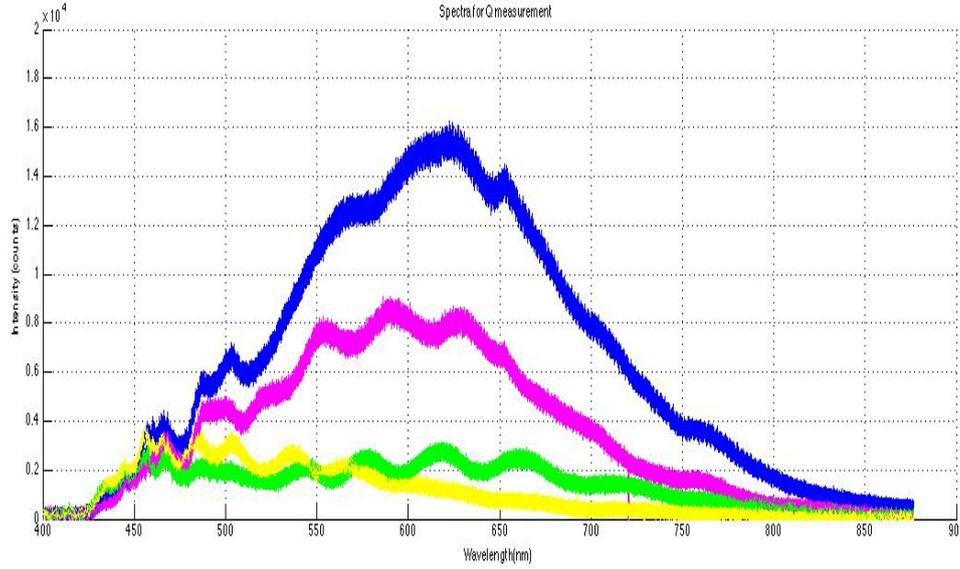
Fig.24 Q measurement

Each spectrum corresponds to one modulation state of the 2 FLCs in use. In order to separate the 4 spectra to proceed to the modulation process we put a threshold at 620 nm where all four spectra can be easily distinguished.

4.2 Modulation Process

After taking measurements for ± Q and ± U we can now form a 4x4 matrix of our measurements: (index 1 means 1st modulation state so for Q measurement we have Q_1, Q_2, Q_3, Q_4 4 spectra for 4 modulation states)

$$I_m = \begin{pmatrix} Q_1 & -Q_1 & U_1 & -U_1 \\ Q_2 & -Q_2 & U_2 & -U_2 \\ Q_3 & -Q_3 & U_3 & -U_3 \\ Q_4 & -Q_4 & U_4 & -U_4 \end{pmatrix}$$

In order to calculate the modulation matrix we follow Eq.(14)

$$\begin{pmatrix} Q_1 & -Q_1 & U_1 & -U_1 \\ Q_2 & -Q_2 & U_2 & -U_2 \\ Q_3 & -Q_3 & U_3 & -U_3 \\ Q_4 & -Q_4 & U_4 & -U_4 \end{pmatrix} = \begin{pmatrix} O_{11} & O_{12} & O_{13} & O_{14} \\ O_{21} & O_{22} & O_{23} & O_{24} \\ O_{31} & O_{32} & O_{33} & O_{34} \\ O_{41} & O_{42} & O_{43} & O_{44} \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Inverting the previous equation we can obtain the Modulation Matrix from the measurements as presented in Fig.25
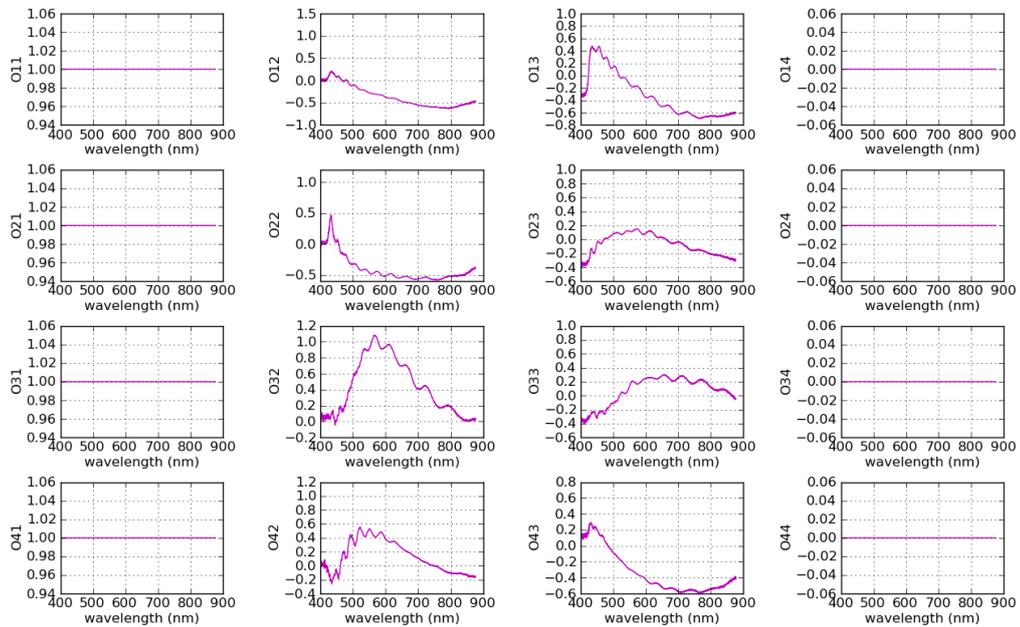
Fig.25 Experiment's Modulation Matrix Elements with wavelength

In order to check the efficiency of our system we calculated the polarimetric efficiencies by following the methodology described above [6](see §1.2.2). The result of this method is in Fig.26
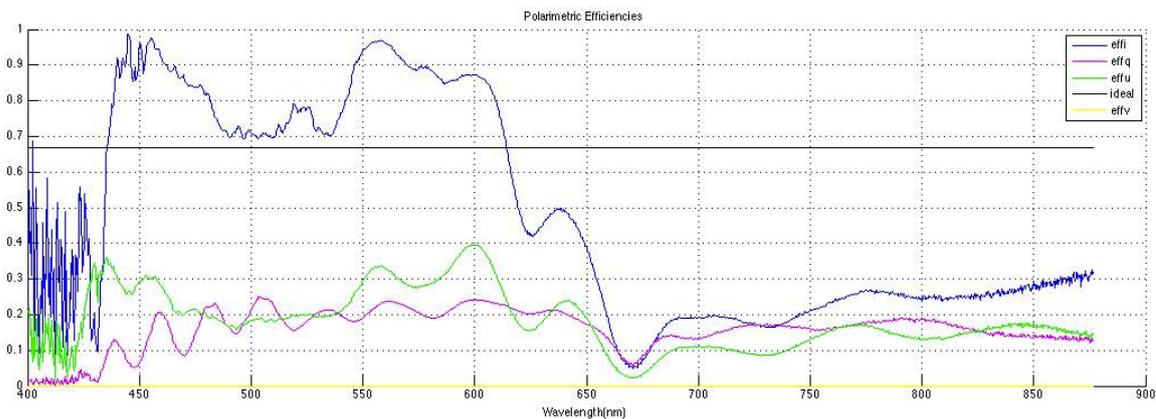
Fig.26 Polarimetric Efficiencies from the Measurements

Normally the efficiencies resulted from the measurements should be in accordance with the efficiencies calculated with the numerical algorithm we run when building the instrument. Unfortunately, this is not the case. So while the efficiency for the Intensity (effi) should approach the value 1 for all the wavelength range, we see a large deviation and stong dependence on the wavelength.

4.3   Comparison with Model

After having the results from the model and our measurements we have the best possible agreement with the model.
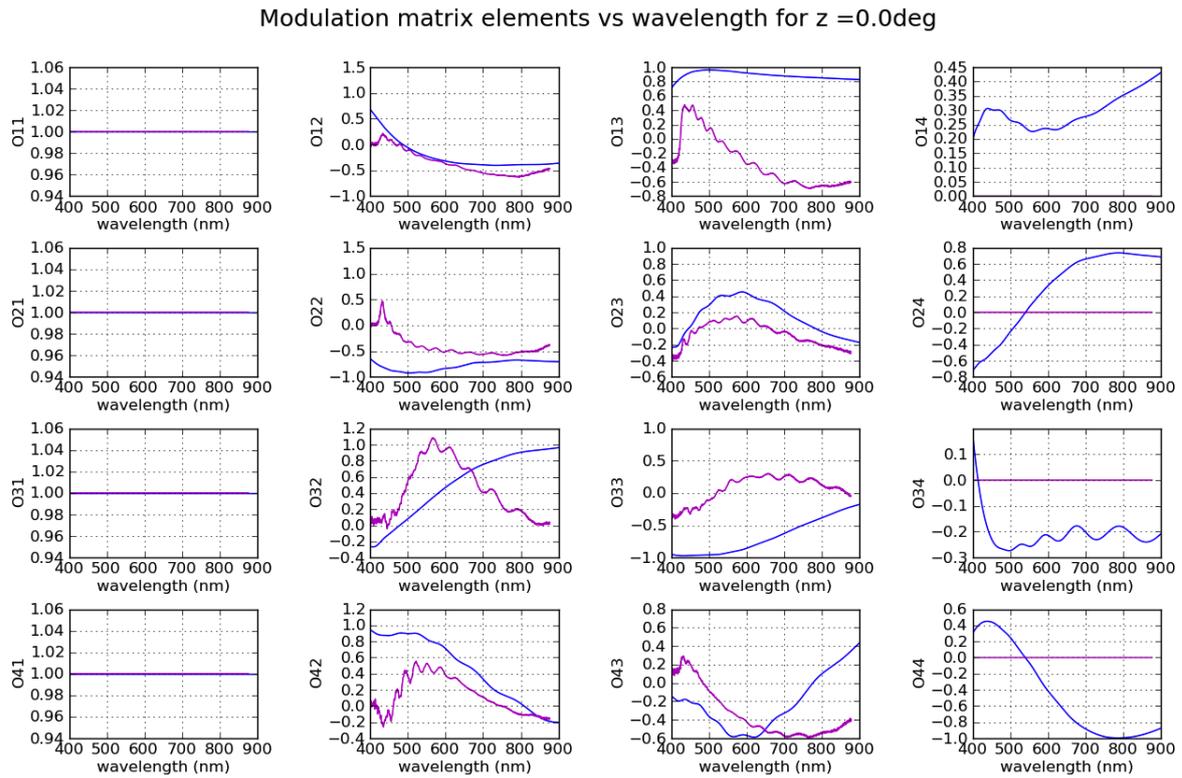
Fig.27 Best agreement between the measurements and the model

R. Diamantopoulou
August 29, 2012

Design and Performance Analysis
of the Poly-Polarimeter

Chap 5: Conclusion and
Recommendations

5 Outlook And Recommendations

A polychromatic modulator can be a very complicated system to work with. There were many unpredicted factors that affected the result and there are more to be determined in a next research. The FLC is a very delicate element but its great advantage is its high speed which enabled us to build such an instrument and achieve the same basic conditions for our sets of measurements. The fast FLCs were the reason of a good synchronisation between the system and the spectrograph.

As we can easily observe from the previous plots the agreement between the model and the measurements was not exactly as we expected it to be. Maybe, the full version of the Poly-polarimeter proves to be more stable. The fact that it was the first time we reacted with FLCs in the lab might have caused some additional deviations from the predicted behaviour. Maybe a first approach to solve this problem could be to build a model which fully simulates the behaviour of FLCs,including the errors from the different parameters which cause deviations from the ideal behaviour. Moreover,the slightest misalignment or other external factors could have caused problematic behaviour in the system.

Another significant parameter that needs to be taken into consideration in the future, is the separation of the 4 spectra for each measurement. The data reduction technique to actually distinguish the spectra for the 4 different modulation states it is not as trivial as it looks like. Thus, we might have ended up mixing the modulation states or even mixing the order of the modulation states when forming the 4x4 matrix that represents the measured intensities.

In the end we managed to build a polychromatic modulator and be able to take actual measurements. We characterise the behaviour of the FLCs, measuring the retardance dependence on wavelength and the partial polarisation due to the polarised fringes. The modulator is already mounted in the lab and we have a perfectly well functioning synchronisation box for all 4 FLCs (to build the whole ellipsometer).

So in conclusion we manage to build a generic lab instrument that can have a lot of applications. It can be used for example for SPEX calibration or even for measuring samples using the M&m's code.

BIBLIOGRAPHY

[1] F.Snik and C.U.Keller "Astronomical polarimetry;polarized views of stars and planets", In T.D.Oswalt, H.Bond, and Others, editors, Planets, Stars & Stellar Systems Springer,2012

[2] http://www.phys.hawaii.edu/ anita/new/papers/militaryHandbook/polariza.pdf

[3] "Astrophysical Spectropolarimetry" Edited by J.Trujillo-Bueno, Cambridge University Press

[4] Jaap Tinbergen ," Astronomical Polarimetry"

[5] Jose Carlos Del Toro Iniesta ," Introduction to Astronomical Polarimetry"

[6] Del Toro Iniesta & Collados, ," Optimum Modulation and demodulation matrices for Solar polarimetry", Optical Society of America,2000

[7] Tomczyk, StevenCasini, RobertoWijn, Alfred G De Nelson, Peter G ," Wavelength-Diverse Polarization Modulators for Stokes Polarimetry" , 2010

[8] T.W.Jones and S.L.O'Dell ," Physical Conditions in Polarized Compact Radio Sources" , 1977

[9] Gerard van Garten ," Development of a precision ellipsometer" , 2008

[10] Keller ," Recent progress in imaging polarimetry" , Solar physics 164-242, 1996

[11] Maria de Juan Ovelar," M & m s : An error budget and performance simulator code for polarimetric systems" , 2011

[12] Gisler Daniel, Aex Feller, Achim Gandofer," Achromatic Liquid crystal polarization modulator" , 2011

[13] Xu, Peizhi Li, Xihua Muravski, Anatoli Chigrinov, Vladimir," P-169 : Photoaligned Bistable FLC Displays with Birefringent Color Switchingr" , 2006

[14] Snik F." Design for Ba II 4554 / H? imaging polarimetry with the Dutch Open Telescope" , 2005

[15] Keller, C.U ." Instrumentation for astrophysical spectropolarimetry" , 2002

[16] Keller, C.U , Snik, F." Polarimetry from the Ground-up" , 2010

[17] Compain, E Poirier, S "General and self-consistent method for the calibration of polarization modulators, polarimeters, and mueller-matrix ellipsometers." , 1999

[18] Ichimoto, K. Lites, B. Elmore, D. Suematsu, Y. et.al "Polarization Calibration of the Solar Optical Telescope on board Hinode" , 2008

APPENDIX A

Photon Spectral Calculation

For the detector:

1. With the Sony 2048 detector we get 40 photons per count at 600 nm (40/count @ 600 nm) with the maximum number of counts $2^{16} = 65536$.

2. So in order to calculate the number of photons at saturation we have to multiply the number/count and the maximum number of counts. In other words 40000.

3. There is a factor of 10 between saturation intensity and lowest intensity in spectrum so the minimum amount of photons at saturation would be 262144.

4. An accuracy of 10^{-4} order which is translated to 10^8 photons/pixel.

5. The number of saturated shots required for this efficiency is $10^8/262144 = 382$ shots.

6. The maximum speed of a spectrograph is 250 Hz, thus the total integration time is 382/250=1.53 sec so in this case we need a light source that provides $10^8 photons/1.53 = 6.54 * 10^7$ photons/sec/pixel.

7. 1650 pixels for 350-1100 nm $\frac{1100-350}{1650} = 0.45$ nm/pixel.

8. The light source should provide $6.54 * 10^7/0.45 = 1.45 * 10^8$ photons/sec/nm.

9. The maximum energy is $E_{max} = \frac{hc}{\lambda} \rightarrow E_{max} = hc/350 * 10^{-9} = 5.68 * 10^{-19}$ so in terms of energy the light source should provide $1.45 * 10^8 * 5.68 * 10^{-19} = 8.24 * 10^{-11}$ W/nm

10. System's throughput plus losses from external factors is almost 1%

11. So in the end the light source should provide $8.24 * 10^{-9}$ W/m

12. For the Ocean Optics DH-2000 light source with minimum output at 350 nm with an energy of $100 * 10^{-6} W/cm^2/nm$

13. The fiber area at $400 \mu m$ is $A = \pi * 0.04^2 = 5.03 * 10^{-3} cm^2$ so the final output of a DH-2000 light source would be $500 * 10^{-9} W/nm$ which meets the requirement of $\geq 8.24 * 10^{-9} W/nm$

The total calculation of photons for the imaging polarimeter (with and without filter) is also included in the next table, as well as the calculation for the photons needed with the detector as reference.

Imaging with DH-2000		Spectrograph	Avantes AvaSpec-1650F-2-USB2
No Filter	10 nm wide filter @350 nm (worst scenario)		Detector-Sony 2048

Imaging with DH-2000 (No Filter)	Imaging with DH-2000 (10 nm wide filter @350 nm)	Spectrograph / Detector-Sony 2048
✓ Ocean Optics DH-2000 light total spectral output @350-900 nm: $800*(900-350)=440,000\ \mu W/cm^2$	✓ Ocean Optics DH-2000 light source with minimum output at 350 nm with an energy of $100*10^{-6}\ W/cm^2/nm$	✓ 40 photons/count @ 600 nm with the maximum number of counts $2^{16}= 65536$.
✓ $A = \pi*0.04^2 = 5.03*10^{-3}\ cm^2$ so the final output of a DH-2000 light source $2200\ \mu W$	✓ $A = \pi*0.04^2 = 5.03*10^{-3}\ cm^2$ so the final output of a DH-2000 light source would be $500*10^{-9}\ W/nm$	✓ Number of photons at saturation : $40*2^{16} = 2621440$.
	✓ 10 nm wide filter:DH-2000 output:$5*10^{-6}$ W	✓ a factor of 10 between saturation intensity and lowest intensity in spectrum so the minimum amount of photons at saturation would be 262144.
✓ System's throughput plus losses from external factors is almost 1%	✓ System's throughput plus losses from external factors is almost 1%	✓ An accuracy of 10^{-4} order which is translated to 10^8 photons/pixel.
✓ Power reaching detector:$2.2*10^{-5}$ W	✓ Power reaching detector:$5*10^{-9}$ W	✓ The number of saturated shots required for this efficiency is 108/262144 = 382 shots.
✓ $E = hc/\lambda \rightarrow Emax = hc/625*10^{-9} =3.18*10^{-19}\ J$	✓ $E = hc/\lambda \rightarrow Emax = hc/350*10^{-9} =5.68*10^{-19}\ J$	✓ Maximum speed of a spectrograph is 250Hz ,thus the total integration time is 382/250=1.53 sec so in this case we need a light source that provides 108 photons/1.53 = $6.54*10^7$ photons/sec/pixel.
✓ #photons/sec= $2.2*10^{-5}$ /$3.18*10^{-19}$=$6.92*10^{13}$ J (how fast should the camera run)	✓ #photons/sec= $5*10^{-9}$ /$5.68*10^{-19}$=$8.80*10^9$ J (how fast should the camera run)	✓ 1650 pixels for 350-1100 nm (1100−350)/1650 = 0.45 nm/pixel.
✓ #photons/sec/pixel= $6.92*10^{13}$ / $(640*480)= 2.25*10^8$	✓ #photons/sec/pixel= $8.80*10^9$ / $(640*480)= 2.86*10^4$	✓ The light source should provide $6.54*10^7$/0.45 = $1.45*10^8$ photons/sec/nm.
✓ QE ~ 0.25 , #e⁻ /sec/pixel=$5.63*10^7$	✓ QE @350 nm: 0.05	✓ The maximum energy is Emax = hc/λ $\rightarrow$ Emax = hc/350 * 10^{-9} =$5.68*10^{-19}$ so in terms of energy the light source should provide $1.45*10^8 * 5.68*10^{-19} = 8.24*10^{-11}$ W/nm
	✓ #photons/sec/pixel=1430	✓ System's throughput plus losses from external factors is almost 1%
✓ Full well depth:30000 e⁻	✓ Full well depth:13000 e⁻	✓ So in the end the light source should provide $8.24*10^{-9}$ W/m
✓ Detector saturated after $5*10^{-4}$ sec camera frequency~2 kHz!!!!!	✓ Detector saturated after 9 sec	✓ Ocean Optics DH-2000 light source with minimum output at 350 nm with an energy of $100*10^{-6}$ W/cm²/nm
✓ Our camera will have ~100Hz (we can throw away a factor of 20 of light using color filters, without loosing accuracy)	✓ Integration time for 10^8 photons :10^8 / 1430= 70000 sec, whereas for 10^4 photons:10^4 /1430 = 7 sec	✓ The fiber area at 400μm is A = $\pi*0.04^2 = 5.03*10^{-3}$ cm² so the final output of a DH-2000 light source would be $500*10^{-9}$ W/nm which meets the requirement of $\geq 8.24*10^{-9}$ W/nm

Fig.A1 Light Source DH-2000 power from 200-1000nm

APPENDIX B

■ **Polarization State Generator**

Lightsource
WP1   ; λ/4 @ 670 nm   ; 161.102°
FLC0   ; λ/2 @ 491.25 nm   ; 51.1299° & 50.1299-44.75=5.3799°
WP2   ; λ/4 @ 488 nm   ; 106.543°
FLC3   ; λ/2 @ 495.00 nm   ; ± 44.75°/2
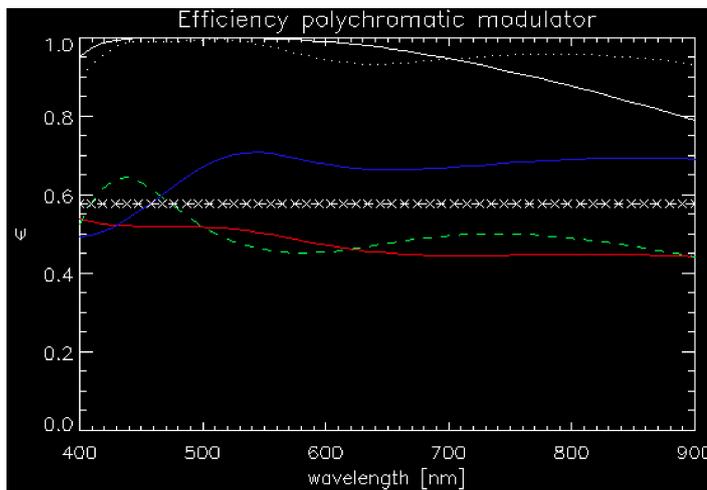
■ **Polarization State Analyzer**

FLC2   ; λ/2 @ 496.50 nm   ; ± 44.25 / 2°
WP3   ; λ/4 @ 488 nm   ; 106.685°
FLC1   ; λ/2 @ 495.00 nm   ; 51.0129° & 51.0129-44.00=7.0129°
WP4   ; λ/4 @ 670 nm   ; 161.204°
Detector
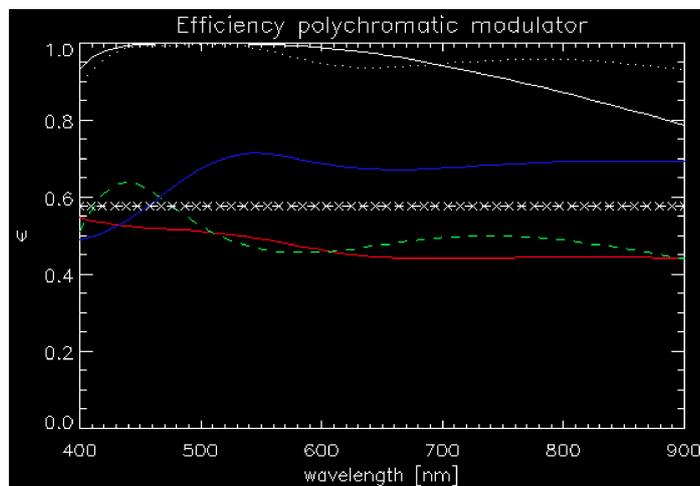
APPENDIX C (NOTE: The original code belongs to Maria de Juan Ovelar. This is just a modified version for the specific project)

1 · ModulModel.py · 2012-08-16 16:12 · Renia Diamantopoulou

```python
#!/usr/bin/env python2.6
# -*- coding: utf-8 -*-

"""
Error Budget calculator for a modulator consisting of (POL, WP, FLC, WP, FLC, POL)
TEST vesion
system: Poly-polarimeter's Modulator
date: June/July 2012
author: R.Diamantopoulou <rdiamant2011@gmail.com>, Leiden University
"""


import os
import sys
import numpy as np
import matplotlib as mpl
import cPickle as cP
import time as t

import Mnms_new
from Mnms_new import *

#======================== INPUT =============================================#
#=============#
# Output file #
#=============#

file = str(raw_input("Output filename? :")) + '.pkl'

A = t.time()


'Wavelength List Definition'
#============#
# Wavelength #
#============#


#list of wavelengths were the modulator is working without malfunctions of the elements
wave_list = np.arange(400., 901., 1.)
#wave = input("Incoming radiation's wavelength (nm)?  ")
#print "Incoming radiation's wavelength ", (wave)


'Zenit angle Definition'
#===================#
# Telescope Rotation #
#    Zenit angle    #
#===================#

# list of values for zenit angle (start, stop(not included), step) in nm.
#zen_list = np.arange(0., 100., 10.)
zen_list = [0]


# interactive mode
#zen = input("Zenit angle (deg) ?  ")

'Temperature'
#===================#
#    Temperature    #
#===================#

T_list = [20.0] # temperature in deg
```

```python
#if you want to test the temperature dependence of the instrument then you add more
values to the T_list

'Modulation'
#===================#
#     Modulation    #
#===================#

mstates = 4 # nr of modulation states
mslist = np.arange(mstates)

#=======================#
# Incoming stokes vector #
# Sinc? I; Q; U; V       #
#=======================#

Sin = np.matrix(raw_input("Incoming Stokes vector (format: I; Q; U; V) ?  "))


#========================================================#

# Creating the dictionary to store the results
# of the simulations
MMM_T = {}

for T in T_list:
    MMM_zen = {}

    for zen in zen_list:
        MMM_wave = {}

        for wave in wave_list:
            MMM_mode = {}
            wavemic = wave/1000.

            for m in mslist:

                # modulation scheme
                #switchangle = 57.29047 - 0.42319*T
                #angdev = abs(45.-switchangle)

                if m == 0 :
                    FLCrotangle = 51.1299 +44.75
                    FLC2rotangle = +44.75/2
                elif m == 1 :
                    FLCrotangle = 51.1299
                    FLC2rotangle = +44.75/2
                elif m == 2 :
                    FLCrotangle = 51.1299+44.75
                    FLC2rotangle = -44.75/2
                elif m == 3 :
                    FLCrotangle = 51.1299
                    FLC2rotangle = +44.75/2
                print "#============= m = ",m," =============#"

                #Creating the input,delta p's and delta_str lists

                Input_mat = [ ]
                deltap = [[ ]]
                deltap_str = [[ ]]


#           ========================================================#
#             GENERATOR!!
```

3 · ModulModel.py · 2012-08-16 16:12 · Renia Diamantopoulou

```python
        print '###### GENERATOR #####'

        generator = RPPOL(0., 1., 0.)

        Input_mat.append(generator[0])
        deltap.append(generator[-2])
        deltap_str.append(generator[-1])




    #=======================================================#
    # RETARDER!!!
    #=======================================================#


    #####################
    # WAVE PLATE
    #####################

    print '\n'
    print '###### WAVE PLATE #####'
#
    designwave = 0.510 #um
    bdesign = 0.134523 + (0.00354683/designwave**2) + (0.00012492/
designwave**4)
    b = 0.134523 + (0.00354683/wavemic**2) + (0.00012492/wavemic**4)
    thick = 0.5*designwave/bdesign # =  um
#
    FLCret_1 = WPb(b, thick, wave, 'FLCret_1')
    print 'FLCret_1 = ', range(len(FLCret_1))

    Input_mat.append(FLCret_1[0])
    deltap.append(FLCret_1[-2])
    deltap_str.append(FLCret_1[-1])




#-------------------------------------------------------------------------------------
--------------------------------#

#-------------------------------------------------------------------------------------
--------------------------------#

#-------------------------------------------------------------------------------------
--------------------------------#

#-------------------------------------------------------------------------------------
--------------------------------#

#-------------------------------------------------------------------------------------
--------------------------------#




    #=======================================================#
    #FLC
    #=======================================================#


    ##############################
    # 1st ROTATION BEFORE 1st FLC_11
    ##############################
    print '\n'
```

4 · ModulModel.py · 2012-08-16 16:12 · Renia Diamantopoulou

```python
                print '###### FLC rot in #####'

                FLCrot_11 = ROT(-FLCrotangle)
                print "FLC rotangle = ", -FLCrotangle
                print 'FLCrot_11 = ', range(len(FLCrot_1))

                Input_mat.append(FLCrot_11[0])
                deltap.append(FLCrot_11[-2])
                deltap_str.append(FLCrot_11[-1])

                #######################
                # Retarder IN  first FLC
                #######################


#                print '\n'
#                print '###### FLC Retarder #####'
#
#                designwave = 0.510 #um
#                bdesign = 0.134523 + (0.00354683/designwave**2) + (0.00012492/
designwave**4)
#                b = 0.134523 + (0.00354683/wavemic**2) + (0.00012492/wavemic**4)
#                thick = 0.5*designwave/bdesign # =  um
#
                FLCret_2 = WPb(b, thick, wave, 'FLCret_2')
                print 'FLCret_2 = ', range(len(FLCret_2))

                Input_mat.append(FLCret_2[0])
                deltap.append(FLCret_2[-2])
                deltap_str.append(FLCret_2[-1])

                #######################
                # FLC 1/POLARIZER 1 _11
                #######################
                print '\n'
                print '###### FLCPolarizer #####'


                thickp_11 = thick*0.85
                #thickp = thick*0.42
                no_11 = 1.5
                #nfs =np.sqrt( 1 + 0.6961663*wavemic**2/(wavemic**2-0.0684043**2) +
0.4079426*wavemic**2/(wavemic**2-0.1162414**2) + 0.8974794*wavemic**2/
(wavemic**2-9.896161**2) )
                ne_11 = (b+no_11)
#                Ke = ((ne/1.5)**2+1.)/(2.*(ne/1.5)) #ne/no divided by 1.5 because
#                Ko = ((ne/1.5)**2+1.)/(2.*(no/1.5)) #FLC is in between glass
                arf_11 = 0.05
                gfactor_11 = np.sqrt(no_11+arf_11)
                #gfactor = 1.8
                #f = 10
                Ke_11 = ( (ne_11/gfactor_11)**2+1.)/(2.*(ne_11/gfactor_11) ) #ne/no
divided by 1.5 because
                Ko_11 = ( (no_11/gfactor_11)**2+1.)/(2.*(no_11/gfactor_11) ) #FLC is in
between glass
                be_11 = 2.*np.pi*ne_11/wavemic
                bo_11 = 2.*np.pi*no_11/wavemic
                fo_11 = bo_11*thickp_11
                fe_11 = be_11*thickp_11
                px_11 = np.sqrt( 1./ ( np.cos(fo_11)**2.+(Ko_11**2)*np.sin(fo_11)**2 ) )
                py_11 = np.sqrt( 1./ ( np.cos(fe_11)**2.+(Ke_11**2)*np.sin(fe_11)**2 ) )

                FLCpol_11 = RPPOL(0., px_11, py_11)
                print 'FLCpol_11 = ', range(len(FLCpol_11))
```

5 · ModulModel.py · 2012-08-16 16:12 · Renia Diamantopoulou

```python
                Input_mat.append(FLCpol_11[0])
                deltap.append(FLCpol_11[-2])
                deltap_str.append(FLCpol_11[-1])


                ########################
                # FLC 1/POLARIZER 2 _12
                ########################
                print '\n'
                print '###### FLCPolarizer ######'

                thickp_12 = thick*0.25 #the thin polariser
                #thickp = thick*0.42
                no_12 = 1.5
                #nfs =np.sqrt( 1 + 0.6961663*wavemic**2/(wavemic**2-0.0684043**2) +
0.4079426*wavemic**2/(wavemic**2-0.1162414**2) + 0.8974794*wavemic**2/
(wavemic**2-9.896161**2) )
                ne_12 = (b+no_12)
#                 Ke = ((ne/1.5)**2+1.)/(2.*(ne/1.5)) #ne/no divided by 1.5 because
#                  Ko = ((ne/1.5)**2+1.)/(2.*(no/1.5)) #FLC is in between glass
                arf_12 = 0.15
                gfactor_12 = np.sqrt(no_12+arf_12)
                #gfactor = 1.8
                #f = 10
                Ke_12 = ( (ne_12/gfactor_12)**2+1.)/(2.*(ne_12/gfactor_12) ) #ne/no
divided by 1.5 because
                Ko_12 = ( (no_12/gfactor_12)**2+1.)/(2.*(no_12/gfactor_12) ) #FLC is in
between glass
                be_12 = 2.*np.pi*ne_12/wavemic
                bo_12 = 2.*np.pi*no_12/wavemic
                fo_12 = bo_12*thickp_12
                fe_12 = be_12*thickp_12
                px_12 = np.sqrt( 1./ ( np.cos(fo_12)**2.+(Ko_12**2)*np.sin(fo_12)**2 ) )
                py_12 = np.sqrt( 1./ ( np.cos(fe_12)**2.+(Ke_12**2)*np.sin(fe_12)**2 ) )

                FLCpol_12 = RPPOL(0., px_12, py_12)
                print 'FLCpol_12 = ', range(len(FLCpol_12))

                Input_mat.append(FLCpol_12[0])
                deltap.append(FLCpol_12[-2])
                deltap_str.append(FLCpol_12[-1])


                ##########################
                # 2nd ROTATION AFTER 1st FLC
                ##########################
                print '\n'
                print '###### FLC rot out ######'
                FLCrotback_21 = ROT(FLCrotangle)
                print "FLC rotangle = ", FLCrotangle
                print 'FLCrotback = ', range(len(FLCrotback))

                Input_mat.append(FLCrotback_21[0])
                deltap.append(FLCrotback_21[-2])
                deltap_str.append(FLCrotback_21[-1])


    #--------------------------------------------------------------------------------------
    -------------------------------#

    #--------------------------------------------------------------------------------------
    -------------------------------#
```

6 · ModulModel.py · 2012-08-16 16:12 · Renia Diamantopoulou

```python
#------------------------------------------------------------------------------------------
--------------------------------#

#------------------------------------------------------------------------------------------
--------------------------------#

#------------------------------------------------------------------------------------------
--------------------------------#




        #========================================================#
        # RETARDER!!!
        #========================================================#


            #####################
            # WAVE PLATE ret_3
            #####################

            print '\n'
            print '###### WAVE PLATE ######'
#
            designwave = 0.510 #um
            bdesign = 0.134523 + (0.00354683/designwave**2) + (0.00012492/
designwave**4)

            b = 0.134523 + (0.00354683/wavemic**2) + (0.00012492/wavemic**4)
            thick = 0.5*designwave/bdesign # =   um
#
            FLCret_3 = WPb(b, thick, wave, 'FLCret_1')
            print 'FLCret_1 = ', range(len(FLCret_1))

            Input_mat.append(FLCret_3[0])
            deltap.append(FLCret_3[-2])
            deltap_str.append(FLCret_3[-1])




    #------------------------------------------------------------------------------------------
    --------------------------------#

    #------------------------------------------------------------------------------------------
    --------------------------------#

    #------------------------------------------------------------------------------------------
    --------------------------------#

    #------------------------------------------------------------------------------------------
    --------------------------------#

    #------------------------------------------------------------------------------------------
    --------------------------------#




        #========================================================#
        #FLC nr.2
        #========================================================#


            ###############################
            # 1st ROTATION BEFORE 2nd FLC_12
```

7 · ModulModel.py · 2012-08-16 16:12 · Renia Diamantopoulou

```
                ###############################
                print '\n'
                print '###### FLC rot in #####'

                FLCrot_12 = ROT(-FLCrotangle)
                print "FLC rotangle = ", -FLCrotangle
                print 'FLCrot_12 = ', range(len(FLCrot_12))

                Input_mat.append(FLCrot_12[0])
                deltap.append(FLCrot_12[-2])
                deltap_str.append(FLCrot_12[-1])

                #######################
                # Retarder IN  first FLC
                #######################


#                print '\n'
#                print '###### FLC Retarder #####'
#
#                designwave = 0.510 #um
#                bdesign = 0.134523 + (0.00354683/designwave**2) + (0.00012492/
designwave**4)
#                b = 0.134523 + (0.00354683/wavemic**2) + (0.00012492/wavemic**4)
#                thick = 0.5*designwave/bdesign # =  um
#
                FLCret_3 = WPb(b, thick, wave, 'FLCret_3')
                print 'FLCret_3 = ', range(len(FLCret_3))

                Input_mat.append(FLCret_3[0])
                deltap.append(FLCret_3[-2])
                deltap_str.append(FLCret_3[-1])

                #######################
                # FLC 2/POLARIZER 1 _21
                #######################
                print '\n'
                print '###### FLCPolarizer #####'


                thickp_21 = thick*0.85
                #thickp = thick*0.42
                no_21 = 1.5
                #nfs =np.sqrt( 1 + 0.6961663*wavemic**2/(wavemic**2-0.0684043**2) +
0.4079426*wavemic**2/(wavemic**2-0.1162414**2) + 0.8974794*wavemic**2/
(wavemic**2-9.896161**2) )
                ne_21 = (b+no_21)
#                 Ke = ((ne/1.5)**2+1.)/(2.*(ne/1.5)) #ne/no divided by 1.5 because
#                 Ko = ((ne/1.5)**2+1.)/(2.*(no/1.5)) #FLC is in between glass
                arf_21 = 0.05
                gfactor_21 = np.sqrt(no_21+arf_21)
                #gfactor = 1.8
                #f = 10
                Ke_21 = ( (ne_21/gfactor_21)**2+1.)/(2.*(ne_21/gfactor_21) ) #ne/no
divided by 1.5 because
                Ko_21 = ( (no_11/gfactor_21)**2+1.)/(2.*(no_11/gfactor_21) ) #FLC is in
between glass
                be_21 = 2.*np.pi*ne_21/wavemic
                bo_21 = 2.*np.pi*no_21/wavemic
                fo_21 = bo_21*thickp_21
                fe_21 = be_21*thickp_21
                px_21 = np.sqrt( 1./ ( np.cos(fo_21)**2.+(Ko_21**2)*np.sin(fo_21)**2 ) )
                py_21 = np.sqrt( 1./ ( np.cos(fe_21)**2.+(Ke_21**2)*np.sin(fe_21)**2 ) )
```

```python
            FLCpol_21 = RPPOL(0., px_21, py_21)
            print 'FLCpol_21 = ', range(len(FLCpol_21))

            Input_mat.append(FLCpol_21[0])
            deltap.append(FLCpol_21[-2])
            deltap_str.append(FLCpol_21[-1])


            ########################
            # FLC 2/POLARIZER 2 _22
            ########################
            print '\n'
            print '###### FLCPolarizer #####'

            thickp_22 = thick*0.25 #the thin polariser
            #thickp = thick*0.42
            no_22 = 1.5
            #nfs =np.sqrt( 1 + 0.6961663*wavemic**2/(wavemic**2-0.0684043**2) +
0.4079426*wavemic**2/(wavemic**2-0.1162414**2) + 0.8974794*wavemic**2/
(wavemic**2-9.896161**2) )
            ne_22 = (b+no_22)
#              Ke = ((ne/1.5)**2+1.)/(2.*(ne/1.5)) #ne/no divided by 1.5 because
#              Ko = ((ne/1.5)**2+1.)/(2.*(no/1.5)) #FLC is in between glass
            arf_22 = 0.15
            gfactor_22 = np.sqrt(no_22+arf_22)
            #gfactor = 1.8
            #f = 10
            Ke_22 = ( (ne_22/gfactor_22)**2+1.)/(2.*(ne_22/gfactor_22) ) #ne/no
divided by 1.5 because
            Ko_22 = ( (no_22/gfactor_22)**2+1.)/(2.*(no_12/gfactor_22) ) #FLC is in
between glass
            be_22 = 2.*np.pi*ne_22/wavemic
            bo_22 = 2.*np.pi*no_22/wavemic
            fo_22 = bo_22*thickp_22
            fe_22 = be_22*thickp_22
            px_22 = np.sqrt( 1./ ( np.cos(fo_22)**2.+(Ko_22**2)*np.sin(fo_22)**2 ) )
            py_22 = np.sqrt( 1./ ( np.cos(fe_22)**2.+(Ke_22**2)*np.sin(fe_22)**2 ) )

            FLCpol_22 = RPPOL(0., px_22, py_22)
            print 'FLCpol_22 = ', range(len(FLCpol_22))

            Input_mat.append(FLCpol_22[0])
            deltap.append(FLCpol_22[-2])
            deltap_str.append(FLCpol_22[-1])


            ###########################
            # 2nd ROTATION AFTER 2nd FLC
            ###########################
            print '\n'
            print '###### FLC rot out #####'
            FLCrotback_22 = ROT(FLCrotangle)
            print "FLC rotangle = ", FLCrotangle
            print 'FLCrotback = ', range(len(FLCrotback))

            Input_mat.append(FLCrotback_22[0])
            deltap.append(FLCrotback_22[-2])
            deltap_str.append(FLCrotback_22[-1])


    #----------------------------------------------------------------------------------
    -------------------------------#
```

9 · ModulModel.py · 2012-08-16 16:12 · Renia Diamantopoulou

```python
#--------------------------------------------------------------------------------------------
-------------------------------#

#--------------------------------------------------------------------------------------------
-------------------------------#

#--------------------------------------------------------------------------------------------
-------------------------------#

#--------------------------------------------------------------------------------------------
-------------------------------#


        #=========================================================#
        # Analyzer!!
        #=========================================================#

            #####################
            # POLARIZER
            #####################
            print '\n'
            print '###### Analyzer #####'

            Analyzer = RPPOL(0., 1., 0.)

            Input_mat.append(Analyzer[0])
            deltap.append(Analyzer[-2])
            deltap_str.append(Analyzer[-1])

        #=========================================================#
        # output
        #=========================================================#

            ####################################################
            #TOTAL MUELLER M&m's FOR THE SYSTEM @modulation state i
            ####################################################
#              print '########################################'
#              print '#  ModelMuellerMatrix and collapsed matrix  #'
#              print '#     for wavelenght ',wave,'              #'
#              print '########################################', '\n'

            OUTPUT = MMM(Input_mat,deltap,deltap_str)
            Output = OUTPUT[0]
            deltap = OUTPUT[1]
            deltap_str = OUTPUT[2]

            MAIN = Output[0][0]
            delta_weight = []
            deltasnostruct = []

            for e in range(len(Output)):
                if e != 0:
                    elem = []
                    param = range(len(Output[e]))

                    for p in param:
#                        print "+ Main Matrix dependence on", deltap_str[f][e]
#                        printMat(Output[f][e], "%.4f")
#                        print type(Output[f][e]),'\n'
                        error = Output[e][p]*deltap[e-1][p]
                        deltasnostruct.append(error)
                        elem.append(error)
                    delta_weight.append(elem)
```

```python
        #                print "size of delta_weight =", range(len(delta_weight)), '\n'
        #                print "delta ps = ", deltap

#                print '###########################################################'
#                print '## TOTAL Mueller Matrix for the system with default δp values ##'
#                print
'###########################################################', '\n'
                total_matrix = MAIN + sum(deltasnostruct)
                Total_matrix = total_matrix/total_matrix[0,0]
#                printMat(Total_matrix, "%.6f")
#                print type(Total_matrix)


                print '###########################################################'
                print '##             STOKES vector after the system              ##'
                print '###########################################################',
'\n'
                Sout_ideal = MAIN*Sin
                Sout = Total_matrix*Sin
#                printMat(Sout, "%.6f")
#                print type(Sout)


                mode = {m:{'coll_matrix':Total_matrix, 'exp_matrix': Output, 'Sout':Sout,
'Sout_ideal':Sout_ideal,'Input_mat':Input_mat, 'deltap':deltap, 'deltap_str':deltap_str,
'FLCrotangle':FLCrotangle, 'FLCpx': px_11, 'FLCpy': py_11}}
                MMM_mode.update(mode)

            MMM_wave.update({wave:MMM_mode})

        MMM_zen.update({zen:MMM_wave})

    MMM_T.update({T:MMM_zen})
#=========================================================#
# Modulation matrix
#=========================================================#

MSM_T = MSM(MMM_T,mstates,Sin)


# output file
out_file = open(file, 'wb')

# Pickle the output using the highest protocol available.
cP.dump([MMM_T,MSM_T], out_file, -1)

out_file.close()

B = t.time()
SIMtime = B-A
print 'Simulation time = ', SIMtime
```

48

The following function Calculates the Mueller Matrix for a Rotating Partial Linear Polariser

1 · RPPOL.py · 2012-08-17 11:51 · Renia Diamantopoulou

```python
#-*- coding: utf-8 -*-
import os
import math
import numpy as np


#############################
###############################################

def rppol(rotang, k1, k2):
    """
    Calculates the Mueller Matrix for a Rotating
    Partial Linear Polarizer given the following parameters:

    @param [in] rotang is angle of optical axis relative to the +Q axis of the system
    @param [in] k1 is ratio of transmitted light (Iout/Iin) in the direction parallel to
the optical axis
    @param [in] k2 is ratio of transmitted light (Iout/Iin) in the direction
perpendicular to the optical axis

    @returns P is rotating partial polarizer mueller matrix

    The definition of the matrix can be found in Introduction to Spectropolarimetry,
    J.C. del Toro Iniesta, Cambridge University Press, 2003
    """
    theta = np.radians(rotang)
    a = k1**2. + k2**2.
    b = (k1**2. - k2**2.)/a
    c = 2.*k1*k2/a
    co = np.cos(2*theta)
    si = np.sin(2.*theta)

    P =a/2.* np.matrix([\
     [1., b*co, b*si, 0.],\
     [b*co, co**2.+b*si**2, (1.-c)*co*si,0.],\
     [b*si, (1.-c)*co*si, si**2.+c*co**2., 0.],\
     [0., 0., 0., c]\
     ])

    return P

if __name__ == "__main__":
    import sys
    rppol(float(sys.argv[1]), float(sys.argv[2]), float(sys.argv[3]))
```

The following function Calculates the Mueller Matrix for a compound wave plate

1 · compWP.py · 2012-08-24 14:09 · Renia Diamantopoulou

```python
# -*- coding: utf-8 -*-

### WAVE PLATE
#######################

# -*- coding: utf-8 -*-

import os
import numpy
from numpy import matrix
import WP
from WP import WP
import ROT
from ROT import ROT
import printmat
from printmat import printMat


#############################

# compWP is a function calculating the Mueller Matrix of a rotating 2 layer compound
# wave plate (in this case queart and MgF2) given the following parameters:

# rot_angle = rotation angle for the compound wave plate (deg)
# alignang = alignment angle between layers, in this case should be 90
# thick1,2 = thicknesse for layer 1 (quartz) and 2 (MgF2)
# wave = wavelenght of the incoming light
# name = name of the element

# OUTPUT:
#[0] = matrix for the element
#[1] = name
#[2] = total retardance of the componud in waves


#################################################
def compWP(rot_ang,alignang, thick1, thick2 , wave, name):

    wmic = wave*1e-3   # wavelength in microns

    ###Start rotation
    #############

    #rot_ang = input("Rotation angle for the wave plate (deg)? ")

    SROT = ROT(-rot_ang)
    print "Start-rotation matrix value"
    printMat(SROT)
    print type(SROT)


    ### first layer: Quartz
    #################

    #no = 1.54431  # input("Index of refraction for the ordinary ray (no)?  ")
    # from http://refractiveindex.info/ (Handbook of Optics)
    no = numpy.sqrt( 1 + 0.663044*wmic**2/(wmic**2-0.060**2) + 0.517852*wmic**2/
(wmic**2-0.106**2) + 0.175912*wmic**2/(wmic**2-0.119**2) + 0.565380*wmic**2/
(wmic**2-8.844**2) + 1.675299*wmic**2/(wmic**2-20.742**2) )


    #ne = 1.55343  # input("Index of refraction for the extraordinary ray (ne)? ")
    # from http://refractiveindex.info/ (Handbook of Optics)
    ne = numpy.sqrt( 1 + 0.665721*wmic**2/(wmic**2-0.060**2) + 0.503511*wmic**2/
```

2 · compWP.py · 2012-08-24 14:09 · Renia Diamantopoulou

```
(wmic**2-0.106**2) + 0.214792*wmic**2/(wmic**2-0.119**2) + 0.539173*wmic**2/
(wmic**2-8.792**2) + 1.807613*wmic**2/(wmic**2-19.70**2) )

    dwp_mic = thick1  # input("Thicknes of the plate (microns)? ")

    wpa = WP(no, ne, dwp_mic, wave, "Quartz")
    #WP[0]: matrix
    #WP[1]: material name
    #WP[2]: retardance in rad

    WPa = wpa[0] #matrix

    print "Values for",wpa[1] ,"layer"
    print "no = ", no
    print "ne = ", ne
    print "thickness = ", dwp_mic, "(microns)"



    ### Alingment between layers
    ########################
    print "missalignment between layers"

    #missal_ang = input("Angle of missalingment between the layers of the wave plate?  ")
    #alignang = 90

    AL = ROT(alignang)
    printMat(AL, "%.4f")


    ### Second layer: MgF2
    ##################

    #no = 1.37774  # input("Index of refraction for the ordinary ray (no)?  ")
    # from http://refractiveindex.info/ (Handbook of Optics)
    no = numpy.sqrt( 1 + 0.48755108*wmic**2/(wmic**2-0.04338408**2) + 0.39875031*wmic**2/
(wmic**2-0.09461442**2) + 2.3120353*wmic**2/(wmic**2-23.793604**2) )

    #ne = 1.38956  # input("Index of refraction for the extraordinary ray (ne)? ")
    # from http://refractiveindex.info/ (Handbook of Optics)
    ne = numpy.sqrt( 1 + 0.41344023*wmic**2/(wmic**2-0.03684262**2) + 0.50497499*wmic**2/
(wmic**2-0.09076162**2) + 2.4904862*wmic**2/(wmic**2-23.771995**2) )

    dwp_mic = thick2  # input("Thicknes of the second layer (microns)? ")

    wpb = WP(no, ne, dwp_mic, wave, "MgF2")

    print "Values for",wpb[1] ,"layer"
    print "no = ", no
    print "ne = ", ne
    print "thickness = ", dwp_mic, "(microns)"

    WPb = wpb[0]


    ### Missalingment
    ################
    print "correction for missalignment between layers"

    #missal_ang = input("Angle of missalingment between the layers of the wave plate?  ")


    CORR_AL = ROT(-alignang)
    print "correction-missalignment matrix nominal value"
```

```
3  ·  compWP.py  ·  2012-08-24 14:09  ·  Renia Diamantopoulou
    printMat(CORR_AL)
    print type(CORR_AL)

    ### End Rotation
    ################

    EROT = ROT(rot_ang)

    print "end-rotation matrix main value"
    printMat(EROT)
    print type(EROT)


    ### Total mueller matrix for wave plate 1
    ##############################

    totaldelta = (wpb[2] - wpa[2])/(2*numpy.pi) # total retardance of the wave plate

    totalWP = EROT*CORR_AL*WPb*AL*WPa*SROT

    print "total phase shift = ", totaldelta, "(waves)"
    print "\n"

    print "total mueller matrix for the wave plate"
    printMat(totalWP)
    print type(totalWP)
    print "\n"

    return totalWP, name, totaldelta


if __name__ == "__main__":
    import sys
    compWP(float(sys.argv[0]), float(sys.argv[1]), float(sys.argv[2]), float(sys.argv
[3]), float(sys.argv[4]), str(sys.argv[5]))
    print sys.argv
##################################
```

The following function Calculates the Mueller Matrix for a Retarder

1 · WPB.py · 2012-08-17 11:53 · Renia Diamantopoulou

```python
#-*- coding: utf-8 -*-

### RETARDER MUELLER MATRIX CALCULATOR ###
######################################################

import os
import numpy as np
from numpy import matrix, pi
import printmat
from printmat import printMat


###########################

# WP is a function calculating the Mueller Matrix of a wave-plate
# given the following parameters:

# no = ordinary index of refraction
# ne = extraordinary index of refraction
# dwp_mic = thickness of the plate in microns
# wave = wavelenght of the incoming light (nm)
# wpmat = string with the name of the plate's material

# OUTPUT:
#[0] = matrix for the wave plate
#[1] = name of the material
#[2] = retardance in radians


###############################################

def WPB(b, dwp_mic, wave, wpmat):

    dwp = dwp_mic*1000  # Thickness in microns -> nm

    A = 2*pi/wave
    print A
    phi = (A)*b*dwp
    phideg = phi*360/(2*pi)
    # retardance in deg
    print "#########################################"
    print "                                         "
    print "Phase shift angle phi (rad)= ", phi
    print "Phase shift angle phi (deg) = ", phideg
    print "order (phase-shift angle = m*(pi/2)) = ", phi*2/pi

#     M = numpy.zeros((4,4), dtype = float)
#
#     M[0,:] = [1,0,0,0]
#     M[:,0] = [1,0,0,0]
#     M[1,:] = [0,1,0,0]
#     M[:,1] = [0,1,0,0]
#     M[2,2] = M[3,3] = np.cos( A*(ne-no)*dwp )
#     M[2,3] = np.sin( A*(ne-no)*dwp )
#     M[3,2] = -(np.sin( A*(ne-no)*dwp ))


    print "#########################################"
    print "                                         "
    M = matrix([[1,0,0,0],[0,1,0,0],[0,0,np.cos( A*b*dwp ), np.sin( A*b*dwp )], [0,0,-
np.sin( A*b*dwp ), np.cos( A*b*dwp )] ])

    #print "Values for", wpmat,"layer"
    #print "bire = ", b
```

2 · WPB.py · 2012−08−17 11:53 · Renia Diamantopoulou

```python
#      print   "ne = ", ne
#print "thickness = ", dwp_mic, "(microns)"

#print "Mueller matrix for ", wpmat
printMat( M, "%.4f")
#print type(M)
#print "\n"

    return M, wpmat, phi


if __name__ == "__main__":
    import sys
    print "#########################################"
    print "                                         "
    WPB(float(sys.argv[1]), float(sys.argv[2]), float(sys.argv[3]), float(sys.argv[4]),
str(sys.argv[5]))
    print sys.argv
#################################
```

The following function Calculates the Mueller Matrix of a Rotation

1 · ROT.py · 2012-08-17 11:55 · Renia Diamantopoulou

```python
import os
import numpy as np
import math
from numpy import zeros
import printmat
from printmat import printMat


def ROT(thetadeg):

    theta = np.radians(thetadeg)
    M = np.mat([[1.,0.,0.,0.],[0.,np.cos(2*theta),np.sin(2*theta),0.],[0.,-np.sin
(2*theta),np.cos(2*theta),0.],[0., 0., 0., 1.]])
    #print M[0]

#    row1 = [1.,0.,0.,0.]
#    row2 = [0.,np.cos(2*theta),np.sin(2*theta),0.]
#    row3 = [0.,-np.sin(2*theta),np.cos(2*theta),0.]
#    row4 = [0., 0., 0., 1.]
#
#    M = np.mat([row1,row2,row3,row4])

    return M

if __name__ == "__main__":
    import sys
    ROT(int(sys.argv[1]))
```