UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

STUDY ON THE LIMITING FACTORS OF A COHERENT OPTICAL FIBER LINK, APPLIED TO PHASE TRANSFER FOR FEMTOSECOND SYNCHRONIZATION

TESIS PARA OPTAR AL GRADO DE DOCTOR EN
INGENIERÍA ELÉCTRICA

JORGE ANDRÉS CASTILLO DÍAZ

PROFESOR GUÍA:
ERNEST MICHAEL

MIEMBROS DE LA COMISIÓN:
DAVID RABANUS
LEONARDO BRONFMAN AGUILÓ
FAUSTO PATRICIO MENA MENA

Este trabajo ha sido parcialmente financiado por:
CONICYT,
a través de Becas Chile, el fondo ALMA-CONICYT (Proyectos 31080020, 31090018 y 31110014), y el programa Fondo Basal (Proyecto PFB 06),
y el Joint Alma Observatory,
proveyendo financiamiento de viaje para el Autor.

SANTIAGO DE CHILE
2013

Resumen

El Gran Conjunto Milimétrico / submilimétrico de Atacama (ALMA) es el observatorio astronómico más grande jamás construido. Se encuentra ubicado en el Llano de Chajnantor, a una altitud de 5.000 metros sobre el nivel del mar en el norte de Chile. Se compone de un conjunto de 66 antenas distribuidas en un área de 15 km de diámetro y que opera entre 35 y 950 GHz, Estas antenas están interconectados en un interferómetro capaz de detectar fuentes de señales débiles[1] procedentes del espacio más profundo y producir imágenes con una resolución angular mejor que el telescopio espacial Hubble. Para lograr este rendimiento de las antenas se tiene que registrar las señales procedentes del cielo con un error de tiempo más pequeño que 38 fs. Esta sincronización se realiza mediante señales ópticas transmitidas desde un edificio central utilizando una red de fibra óptica del tipo usado en telecomunicaciones.

El sistema de transmisión se ve afectado por perturbaciones externas tales como la vibración, y oscilaciones térmicas. Estas perturbaciones además de cambiar directamente la longitud de la fibra, afectan el estado de polarización de la luz que viaja a través de ella. Este cambio de longitud puede ser neutralizado aplicando un esquema de corrección de longitud de línea que utiliza estiradores de fibra. Sin embargo, el cambio de polarización no es fácil de compensar, y su interacción con la dispersión de modo de polarización (PMD)[2], produce una variación de fase adicional. Con el fin de minimizar este efecto, ALMA especificó valores de PMD muy bajos para todos los equipos que se ubican en el camino óptico de la señal que transmite la referencia de fase. Puesto que el sistema de corrección de longitud de línea utiliza estiradores mecánicos, se debe poner especial atención en alinear el estado de polarización de la señal que entra al estirador de fibra con su punto óptimo. Solamente encontrando este punto se puede minimizar el cambio de polarización que se produce al estirar la fibra ubicada en el interior del estirador mecánico.

Este trabajo presenta un algoritmo innovador y eficiente para encontrar el eje óptimo de los estiradores de fibra utilizados en ALMA. Este algoritmo se desarrolló durante la realización de este trabajo, y permite una identificación rápida y precisa del eje óptimo que minimiza el cambio de polarización, comparado con el método original, este algoritmo reduce en un 70% el tiempo promedio de calibración y reduce en un 50% el error de calibración residual. También se hace una revisión de los sistemas de transferencia de referencia de tiempo utilizados en ALMA. Explicando los desafíos que enfrentan y las soluciones tecnológicas implementadas para supéralos. A continuación, se presenta el efecto del PMD, mostrando diferentes técnicas de medición y su compensación. Además este trabajo presenta un conjunto de mediciones de PMD realizadas sobre fotomezcladores, estiradores de fibra, e instalaciones de fibra en el Arreglo Muy Grande (o Very Large Array, administrado por NRAO) y en ALMA.

[1] Por ejemplo puede detectar una señal de 26 micro Jansky $2.6 * 10^{-31}\, {}^{W}\!/_{m^2 Hz}$ en una integración de 5 minutos, usando 50 antenas de 12m, frecuencia del cielo 100GHz, ancho de banda 7.5GHz, con 0.47mm de vapor de agua. fuente ALMA Sensitivity Calculator "http://almascience.eso.org/proposing/sensitivity-calculator".

[2] Un efecto dispersivo que produce una dependencia de la velocidad de propagación con el estado de polarización de la señal óptica que viaja por una fibra óptica.

Abstract

The Atacama Large Millimeter/Submillimeter Array (ALMA) is the largest astronomical observatory ever built. It is located on the Chajnantor plateau, at an altitude of 5000 meters above the sea level in northern Chile. It consists of an array of 66 antennas spread in a 15 km diameter area and operating between 35 and 950 GHz. These antennas are interconnected in an interferometer capable to detect weakest signal sources[3] coming from the deepest space and produce images with an angular resolution better than the Hubble space telescope. To achieve this performance the antennas need to record the signals coming from the sky with a time error smaller than 38 fs; this synchronization is made using optical signals transmitted from a central building via a telecommunication grade fiber optic network.

This transmission system is affected by some external perturbations like vibrations and thermal cycles. These perturbations, besides directly changing the length of the fiber, affect the polarization state of the light traveling across it. The length change can be neutralized implementing a line length correction scheme that uses fiber stretchers, but the polarization change is not easy to compensate, and its interaction with the polarization mode dispersion (PMD)[4] produces extra phase drift. In order to minimize this effect, ALMA specified very low PMD for all the equipment in the optical path of the signal carrying the phase references. Given that the line length correction scheme uses fiber stretchers that mechanically change the length of a long section of the fiber, special attention needs to be put in aligning the input polarization state with the axis of this device that minimizes the polarization change produced when the stretcher moves.

This work presents an innovative optimization algorithm for the fiber stretchers used in ALMA, which was developed during the realization of this work. It allows a fast and accurate identification of this axis of minimal polarization change, compared with the original method; this algorithm produced a 70% reduction of the average calibration time and a 50% reduction of the average residual calibration. A review of the time reference transfer system used in ALMA is also made, explaining the challenges that it faces and the way they are overcome. Then the effect of PMD is presented, showing different PMD compensation and measurement techniques. In addition, this work presents PMD measurements made on ALMA photomixers and fiber stretchers, and of buried fiber installations of the Very Large Array (VLA) and ALMA.

[3] As example, it can detect an 26 micro Jansky ($2.6 * 10^{-31} W/_{m^2 Hz}$) signal in an integration time of 5 minutes using 50 12m antennas, at 100 GHz sky frequency, 7.5 GHz bandwidth, with 0.47mm water vapor. Source, ALMA Sensitivity Calculator, "http://almascience.eso.org/ proposing/sensitivity-calculator".

[4] A dispersive effect, that produces propagation speed dependence with the state of polarization of the optical signal that travel in a fiber optics.

This thesis is dedicated to Martin and
Javier, my two sons and to Hilda, my wife;
you are my family and gave me the energy
that I needed to finish this venture.

I love you.

Acknowledgements

First I want to thank Javier Ruiz-del-Solar for his support during the first year of this always challenging academic venture. Then to Jorge May and Leonardo Bronfman, for pushing the idea of an astronomical instrumentation area in this university, you helped me to be in the place where I wanted to be. Then I want to thank Mrs. Eliana Monardes for her diligent actions that made my life a lot easier during my grad school time.

Finally I want to thank Ernest Michael, William Shillue and Patricio Mena for their mentoring and patient advice that allowed me to complete this thesis.

Contents

1 Introduction

1.1 General Motivation

There is an important number of applications that require to transfer frequency and phasing standards. They include the test of fundamental physical principles, development of next-generation accelerator-based x-ray sources, and long-baseline coherent radio telescope arrays, to name a few important examples. In the case of radio telescopes the driving application is the never ending need of astronomers to enhance the angular resolution of their observations. This goal is complicated by the fact that the wavelengths of signals at radio frequencies are sub millimeteric and longer. At these wavelengths the antenna size required to obtain a resolution similar to a backyard optical telescope can be kilometers. Currently, the biggest movable antenna of the world is the Green Bank Telescope and it is only 100-m wide because increasing technical difficulties when building a bigger antenna turns them impractical. Then, the only practical way to increase the angular resolution is the construction of interferometric arrays of smaller antennas. These kinds of arrays are capable of a maximum resolution equivalent to an antenna which diameter is equal to the longest distance between antennas within the array.

To make the interferomentric combination of multiple sub-millimeter antennas it is necessary, as first step, to make a synchronized heterodyne down conversion of the signals reaching the antennas. To understand the need for synchronization we have to consider that the main function and capacity of any interferometer is the precise measurement of phase differences between two signals of the same frequency. Then, any pre-processing of the signals, before the final cross-correlation must have special care in not distorting their phase. This requirement is impossible without adequate time synchronization.

To satisfy this requirement, a precise frequency reference, i.e. a timing signal, must be distributed to the different antennas of the array. A good approach for doing that is to use high-standard telecommunication fibers to send a laser signal containing a phase and frequency reference to the local oscillator system of the receiver. This approach can be as simple as using a single laser with amplitude-modulation. However, it becomes more difficult as the frequency of the reference increases. For frequencies beyond 50 GHz the preferred approach is to use two phase-locked lasers with a frequency difference between them equal to the frequency of the microwave reference that needs to be transmitted to each antenna. Then, an optical mixer down-converts the difference frequency of the two lasers (heterodyning) recovering in this way the microwave reference signal. The largest interferometric array in the world, the Atacama Large Millimeter Array (ALMA) makes use of such a system.

ALMA consists of 66 parabolic antennas spread on a 15-km diameter area at the Chajnantor plateau in northern Chile. Its maximum operating frequency is 950 GHz, thus the requirement of coherency imposed to the First Local-Oscillator (LO) reference signal is an unprecedented phase stability.[5] This degree of accuracy has led to the development of a novel photonic system which involves the active stabilization the optical length of the fiber link by a

[5] equivalent to a maximum time variation of 38 fs $(10^{-15}s)$ in 300s over 15 km of fiber optics, which means a maximum fiber length variation of 8μm. This is an accuracy of 5×10^{-10} for the 15 km fiber.

Line Length Corrector (LLC). This system was designed and built by ALMA engineering groups, and subsequently tested and delivered to the ALMA high-altitude site. The success of this design depends on the quality of the installed fiber optic cable, which includes portions in the ALMA Central Technical Building, buried cable, and cable installed on the antennas themselves. These sections of fiber are all being procured and installed by separate parts of the ALMA project in Chile. Therefore, the performance of the ALMA photonic LO system was not completely characterized, until all of the parts were installed for the first time at the high site. An area of particular concern and interest is the quality of the buried fiber optics cable and it's Polarization Mode Dispersion (PMD).

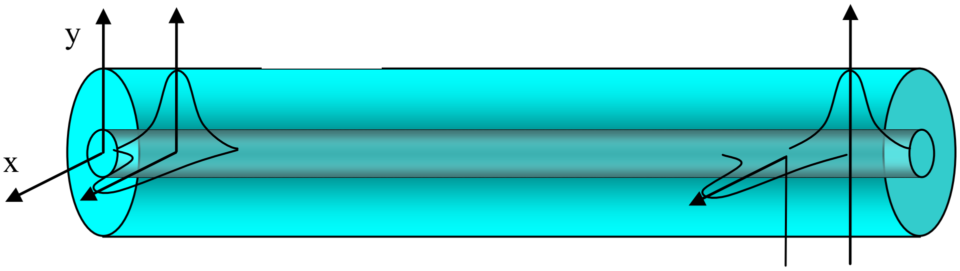


Fig. 1: PMD in a optical fiber. This phenomenon is produced by the intrinsic birefringence of the fiber. As a result the light traveling with a state of polarization parallel to the fast axis of the fiber arrives earlier than the one traveling parallel to the slow axis, producing a phase difference and pulse broadening, at the far end of the fiber.

PMD is produced by the residual, unwanted birefringence of the fiber optics. This implies that the light has different propagation speed depending on its state of polarization, as shown in Fig. 1. Through previous theoretical and experimental work, this is known to be critical to the overall performance of the photonics system, and of the high frequency performance of the array itself.

1.2 Overview of ALMA

ALMA is a huge radio astronomical telescope interferometer installed on Chajnantor plateau in northern Chile. When completed, the facility will consist of an array of up to 54 12-m and 12 7-m parabolic antennas that can detect millimeter and submillimeter wavelength radio waves in the frequency band between 35–950 GHz.

Using interferometric techniques this antenna array will achieve resolutions of 10 milliarcseconds. This resolution is an order of magnitude better than the Hubble telescope or the Very Large Array operating in New Mexico [1].

At the time of writing this thesis, there were more than 50 antennas at the Chajnantor plateau, and the technical facility was completed. The photonics system is a critical component of the telescope that links the different antennas at the high site. Given the extreme coherency specifications of ALMA this system is pushing the coherent link technology beyond its limits. Fig. 2 shows the schematics of the photonic-distribution system.

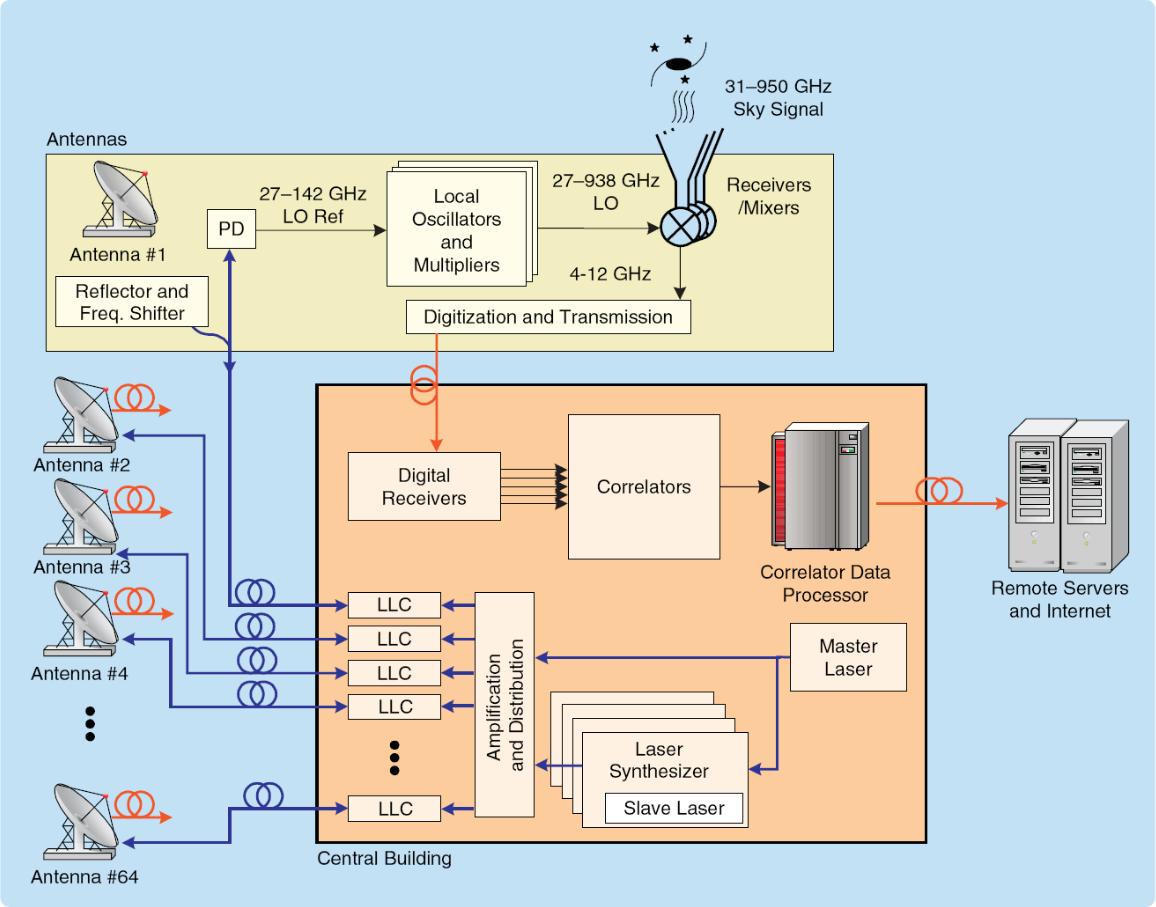


Fig. 2. The photonic local-oscillator reference distribution system. The optical wave from a single master and phase-locked slave lasers are combined in a multi-kilometer optical fiber and sent to each antenna to generate the 27–142 GHz reference signal by photomixing. Each LLC subsystem measures the round-trip length of a fiber using interferometric methods to stabilize the length of fiber to the antenna to within a few micrometers, ensuring accurate timing of the reference signal [1].

The delivery of the ALMA photonic central LO was completed in September 2011, and it is currently in operation at the Chajnantor plateau. The fiber installation has entered into the final stage with the beginning of the implementation of the most remote antenna foundations. This process should be finished by the end of 2013. Fig. **3** shows the actual fiber path at the Chajnantor plateau.

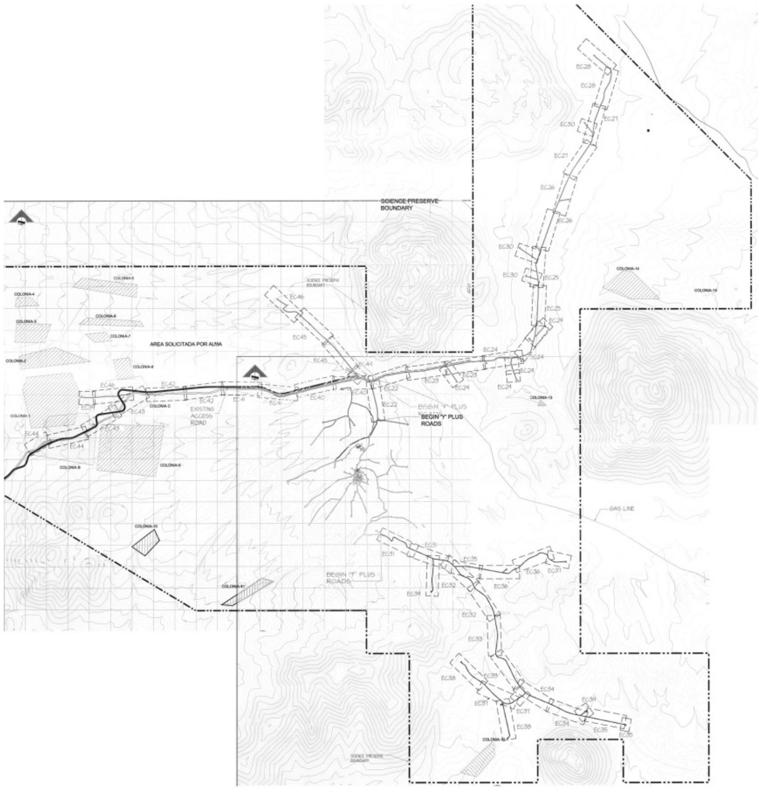


Fig. 3In order to provide photonic local oscillator signal to 192 antenna foundations, a few hundred kilometers of buried fiber optic cables are distributed across the ALMA operations site, their length goes from a few hundred meters and up to 15 km. This diagram shows an overview of the paths of these fibers at the Chajnantor plateau [2], [3], [4].

1.3 Definition of the Problem

Given the photonic nature of the signals transmitted by the ALMA central local oscillator, and the very strict phase stability requirements imposed to the system, the PMD of the optical fiber links that connect the antennas is a parameter that needs to be limited by design, because it can convert any disturbance of the system into phase drifts of the transferred reference frequency signals, and thus, it may limit the performance of the array. At the time when this thesis project was formulated, there was a moderate uncertainty about the final PMD of the fibers installed at the ALMA operations site. However, after the first prospective measurements it was clear that the PMD on the installed fibers was small enough to not present any threat for the phase stability of the array. Indeed, during the measurements the PMD readings where so small, that they were comparable to the measurement error. As a result, the priority given by the ALMA project to this issue was reduced and therefore the funding for a long-term data-collection campaign was reduced to a minimum, so that no further characterization was possible. The results of these first measurements are reported in this document. As consequence, the main research focus of this work was changed to the line length correction scheme used by ALMA. Within this subject the development, implementation, and testing of a faster polarization optimization algorithm for the fiber stretcher of ALMA's line length corrector became the main thesis topic of the author of this document.

1.4 Background

The problem of transferring ultra-stable frequency references via fiber-optic networks has been tackled in literature through three different techniques. These include modulating a continuous wave laser (CWL) with a microwave reference [5], transferring an optical carrier [5], and the transfer of an optical frequency comb [6]. As this thesis will be related to the Local Oscillator (LO) distribution system of ALMA, the following paragraphs will be centered on the first technique which is the one used in ALMA. In this technique the microwave reference is typically used to modulate the intensity of a communication laser, whose signal is transmitted by an optical fiber to the antenna and demodulated by a photo detector. In this way the original microwave signal is recovered. However, given the limitation of the modulators at the time of the project definition, ALMA adopted a slightly different version of this technique that allows the distribution of frequencies up to 142 GHz and more. Rather than modulating a single optical carrier, it is based on phase-locking two lasers with a frequency difference equal to the desired reference frequency. Both optical signals are transmitted through a single optical fiber and, at the antenna, a high frequency photomixer detects the beatnote [7], [8], which is the original microwave signal that drove the phase-lock loop at the beginning. This LO beat note at the antenna can be phase-shifted due to physical changes in the optical line length of the fibers, and by changes in the alignment of the polarization states of the two lasers at the antenna's photonic receiver. This situation degrades the phase characteristics of the reference signal, preventing the system from achieving (i) the phase noise, and (ii) the phase drift[6] specifications of the ALMA LO [8].

1.4.1 Line Length Corrector (LLC)

To overcome the problem of the optical-line-length induced phase drift, it is necessary to stabilize the link in order to correct the phase drift produced by mechanical and thermal perturbations of the fiber. To do so there are two approaches.

(i) One is to measure the phase shift of the microwave signal, the one received from the laser reflection originated on the far end of the link, and then comparing it with the original microwave signal, introducing then a phase offset on the launched signal, in order to compensate the phase drift of the fiber [9][10]. This technique was demonstrated on [9] showing an Allan standard deviation (ASD) of 7.5x10^{-17} on 1000 s over 3 km of installed fibers.

(ii) A different technique makes the phase measurement at optical frequencies (200 THz) [11][12], comparing the phase between a non-modulated light wave and a frequency-shifted round trip copy of it in a fiber-optic Michelson interferometer. This method promises an improvement of orders of magnitude in the measurement precision of the variation in the optical length of the fiber link [13]. It has achieved an ASD of 1.3x10^{-18} on 1000 s over 700 m [6]. The ALMA LLC scheme uses this approach and stabilizes the fiber link with a piezo-driven fiber stretcher [14] as it is illustrated in Fig. 4.

[6] Phase variations with a frequency >1Hz are considered phase noise, otherwise is considered phase drift.

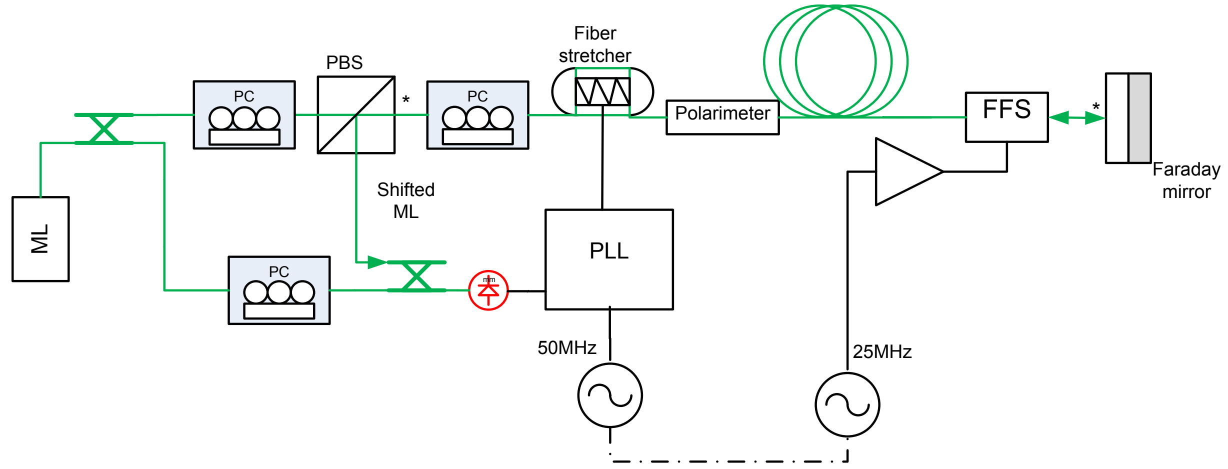


Fig. 4 Simplified diagram of the ALMA line-length correction (LLC) system. A signal generated at the ultra phase-stable master laser (ML) is transmitted to the Faraday rotator mirror, where it is reflected back, thus crossing the fiber frequency shifter (FFS) twice, producing a total frequency shift of 50 MHz and, furthermore, undergoing a Poincaré-sphere inversion of its polarization state. The signal returning to the polarizing beam splitter (PBS) is then mixed with a direct signal from the ML to produce a beat note at 50 MHz. This signal is used to lock the PLL that drives the fiber stretcher, effectively stabilizing the link total length between the asterisks. If a state-of-polarization change (SOPC) of the optical signal is produced by the stretcher, due to the PMD of the rest of the fiber line and other optical components such as the photomixer (not shown in this figure), a phase change of the signal at the antenna will be produced, thus possibly rendering the antenna phase out of the phase-stability specification. Therefore, just before the input of the fiber stretcher, a polarization controller (PC) is inserted which is used together with the polarimeter to find the fiber stretcher sweet-spots [15]**.**

The LLC system is designed to cancel the phase drift produced by variations of the optical-path length of the fiber due to thermal and mechanical changes. The precision with which it does this is usually enormous. A control of the total optical line length of 15 km can keep the phase drift below approximately 10 fs over the standard integration time of 300 s [16]. Fringe slips happen only occasionally due to some very fast and strong perturbations, which cannot be suppressed by these very sensitive, but in this case still slow, fiber stretcher controllers. This slow response is also due to the fact that the correction speed is fundamentally limited by the round trip time of 150 μsec over the fiber length of 15 km. The latest laboratory measurements of microwave phase stability for a LLC-stabilized 14 km wrapped fiber (of the type buried at the ALMA site) indicate that the phase noise is within the ALMA specifications.

Phase fluctuations of the beat signal at the antenna end due to the drift of the polarization are minimal if both polarizations are perfectly matched. As it is shown in references [15] and [17], the amplitude of these phase fluctuations are reduced by increasing the degree of mutual polarization alignment of the two phase-locked light waves that travel through the optical fiber to generate the beat note at the photo mixer. It has been found that this phenomenon is especially important for the fiber stretchers [18][14], given that the fiber in the device suffers large mechanical stress. Thus, to reduce the absolute and relative phase noise associated with polarization drift at the antenna end, the polarization in front of the fiber stretcher is aligned to a principal state of polarization (PSP) of the stretcher, which reduces the state of polarization change (SOPC) while the stretcher moves [19][14]. This is adjusted when necessary using an automatic routine implemented in the LLC firmware or using an external script based on an algorithm developed and explained in this thesis. This algorithm will be presented in detail in chapter 4.

1.4.2 Polarization mode dispersion (PMD)

The PMD is a phenomenon produced by the residual, unwanted birefringence of the optical fiber. Birefringent media have different propagation speed depending on the state of polarization (SOP) of the light traveling through them. These media in general have two propagation modes with orthogonal polarization states (opposite on the Poincaré sphere), known as “principal states of polarization” (PSP) [19]. The so-called fast and slow axis correspond to the two SOPs where all the energy of the light beam travels at the same speed. If the energy is divided between both PSPs there will be a differential of speed between these components which is translated into a phase mismatch of the two components at the end of the transmission medium. This effect is known as differential group delay (DGD) as it is shown in Fig. 1. The DGD is the time difference, Δt, at the end of the fiber.

In a short fiber the DGD will increase linearly with the length, but in a long fiber the different sections of the fiber have their axes aligned randomly, so the laser energy is transmitted by a series on mediums that have a random effect on the signal’s SOP. This situation tends to equalize the DGD. Thus, for long fibers the DGD increases with the square root of the fiber length.

This birefringence comes from some intrinsic factors produced by irregularities in the fiber, and extrinsic factors coming from mechanical and thermal perturbations of the fiber. In long fibers the DGD varies randomly with time following a Maxwellian distribution. In wavelength it also has a random behavior that follows the same probability distribution as it has in time. For an analog transmission system like the ALMA photonic distribution system, the effect of PMD does not degrade the phase noise performance by itself, it only induces a constant phase delay and a fixed power conversion loss penalty at the photomixer. [18].

PMD-related phase drift or jitter occurs through two mechanisms. The first case is when the SOP changes its orientation relative to the PMD vector. Therefore, the power and phase relation of the components in the two PSPs of each of the wavelengths change. This cause the two beat components to have varying amplitude and phase. The result is jittering due to the changing differential SOP between the two wavelengths. As pointed out earlier, the common SOP changes do not contribute to the drift if detected by a perfect receiver.[18] The second case is when the two light-waves are detected by a polarization sensitive detector. The two beat signals are then affected by this polarization sensitivity, and their amplitude and phase will be affected respectively. The overall beat note in this case may not have a constant phase even if the two SOPs rotate in the same way, as their relationship to the PSPs of the detector is changing. In digital communications, where the spectral width of the signals is considered small and thus the SOPs are assumed to rotate together, this effect is referred to as polarization mode noise. For the LO distribution, the polarization mode noise will affect both the phase and amplitude of the delivered beat signal [18].

To reach the goal of supplying a stability better than 12 fs in phase drift and a phase noise RMS better than 38 fs during a measurement period of 1000 s [15], the noise introduced by polarization mode dispersion (PMD) must be very well evaluated and understood . Specifically, the ultra-low temporal drift PMD behavior of a buried fiber at the high site must be evaluated here. Fibers specified to ultra-low DGD-values are of special interest. Additionally, the knowledge of the long-term behavior of the DGD of all eight individual fibers composing an individual fiber link on each antenna pad, must to be characterized [14]. This thesis made contributions on this characterization.

1.4.3 PMD Compensation

There are several methods for equalizing the PMD that can be categorized in two types. One is the optical equalization of the transfer function matrix for an optical fiber with PMD. The second is the optical first-order PMD equalizer, employing a polarization state controller to align the principal states of polarization (PSP) and a delay-time compensator [20].

First order PMD compensation may be very simple. A first approach can be the injection of the signal to be transmitted in one principal state of polarization (PSP), if the input signal is transmitted in just one PSP, the signal will be received in the same way at the end of the fiber. This technique can reduce the PMD effect in the signal. The other approach is to compensate the effect of PMD at the far end of the fiber. One example of this was proposed by Heismann in 1998 [21]. The basic idea here is to separate the two PSP using a polarization controller (PC) and polarizing beam splitter (PBS), to compensate the DGD generated by the fiber and with a variable delay. Then both branches are recombined into one fiber, as shown in Fig. 5. This later approach performs better than the former. There are more sophisticated approaches following the same principle by using fixed delays or including more stages, adding more degrees of freedom to the comparator and thus enabling it to compensate more of the higher order distortions [22]. Fig. 6 shows a schematic description of different techniques.

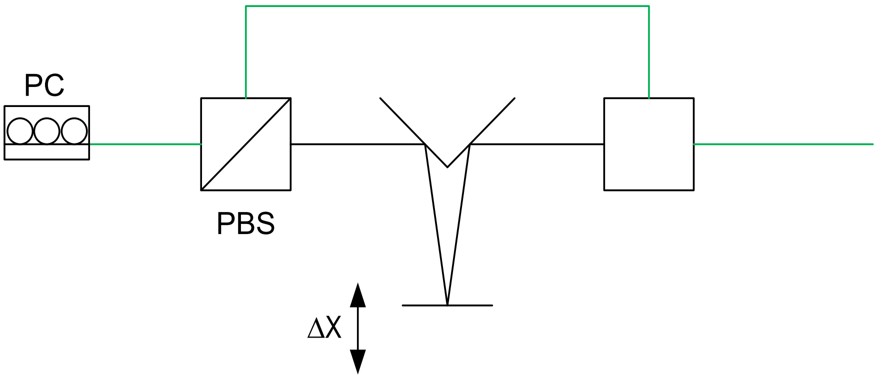


Fig. 5 Simple PMD compensator. A polarization controller (PC) align fast and slow axis of the fiber to the polarizing beam splitter (PBS) then moving mirror adds a variable delay by changing Δx, this can reduce the effect of the DGD of the fiber.

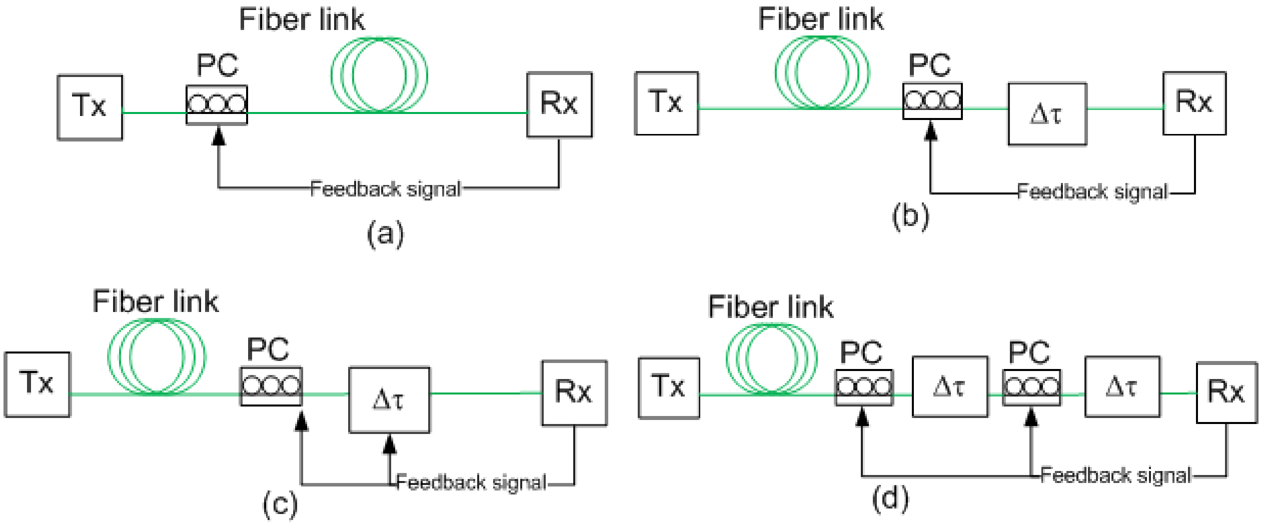


Fig. 6 Compensator schemes. (a) PSP method. (b) Polarization controller (PC)+fixed delay. (c) PC + variable delay. (d) 2 PCs and 2 fixed delays [22]**.**

An important issue when compensating the PMD is the type of feedback signals used to drive the control loops. Some of them rely on polarization scrambling on the input, others simply use a polarizer and measure the average power [23]. The difficulty lies in the multitude of signals and algorithms which must be used to find and track the optimal operation point. There is always risk that the compensator loses track and ends up in an unfavorable position or on a sub-optimal location.

These methods were developed to compensate PMD in digital communication over optical fibers (pulses), so that their application to compensation in CW systems must be tested. In spite of this, the ALMA photonic distribution system adds the particularity of having moving fibers. This movement makes the PMD drift much faster than the typical application and introduces new constraints to the compensator that must be evaluated before use of such compensation techniques in ALMA.

In closing this section, it is necessary to mention that there are other techniques based on bit coding and polarization scrambling associated with bit error correction [24], but these techniques are not applicable to CW systems, so they will not be analyzed here.

1.4.4 Long Term PMD Characterization on Installed Fibers

The PMD is a vector that changes with time in orientation (PSP) and magnitude (DGD). The drift on both orientation and magnitude produce drift in phase of the transmitted signal. This occurs because a change in the DGD causes a change in the relative phase of the components traveling through the fast and slow PSP, then if the orientations of the PSPs change, the distribution of power between both PSPs change. Thus, the manner how PMD changes in time and in frequency will define the performance of an analog signal transmission system like the one studied here.

It has been found that the behavior of PMD has the following characteristics [25]:

- The DGD varies randomly with time and wavelength. The occurrence rate of the DGD-values follows a Maxwellian probability distribution in both domains.

- The average DGD over time equals the one averaged over wavelength.

- The probability distribution of PMD vectors in telecommunication grade fibers is isotropic.

- The variation of PMD vector orientation is correlated with variations of the DGD.

- External mechanical or thermal perturbations have a strong effect on the variation of the PMD.

- The variations of the PMD of two fibers of the same cable are well correlated.

- The DGD varies rapidly over frequency and slowly over time in a fiber protected from external perturbations.

1.4.5 Alternative Line Length Corrector (ALLC)

Given the difficulties encountered during the development of the LLC [26], an alternative scheme for a LLC (ALLC) has been proposed and tested by NAOJ [27][14]. This alternative scheme has a different approach to the problem than the one used by the base line LLC [14].

The baseline LLC measures the phase drift of one fixed laser frequency which makes the round trip through the optical fiber. During this trip the signal is frequency-shifted 50 MHz by an optical frequency shifter. The measurement is made by comparing the original laser signal with this round trip signal on an optical mixer, and then the resulting 50-MHz heterodyne beat note is compared with a locally generated 50 MHz signal.

The ALLC, presented in Fig. 7, makes the heterodyne mixing over the two different laser frequencies generated by the laser synthesizer, comparing the round trip signal with its original copy, and then comparing both 50 MHz beat notes against each other, avoiding the use of a locally generated reference signal and relaxing the requirements of the remote frequency shifter. The result is that the measured phase drift is the one from the microwave signal that has been transmitted rather than the phase of the optical [28] frequencies.

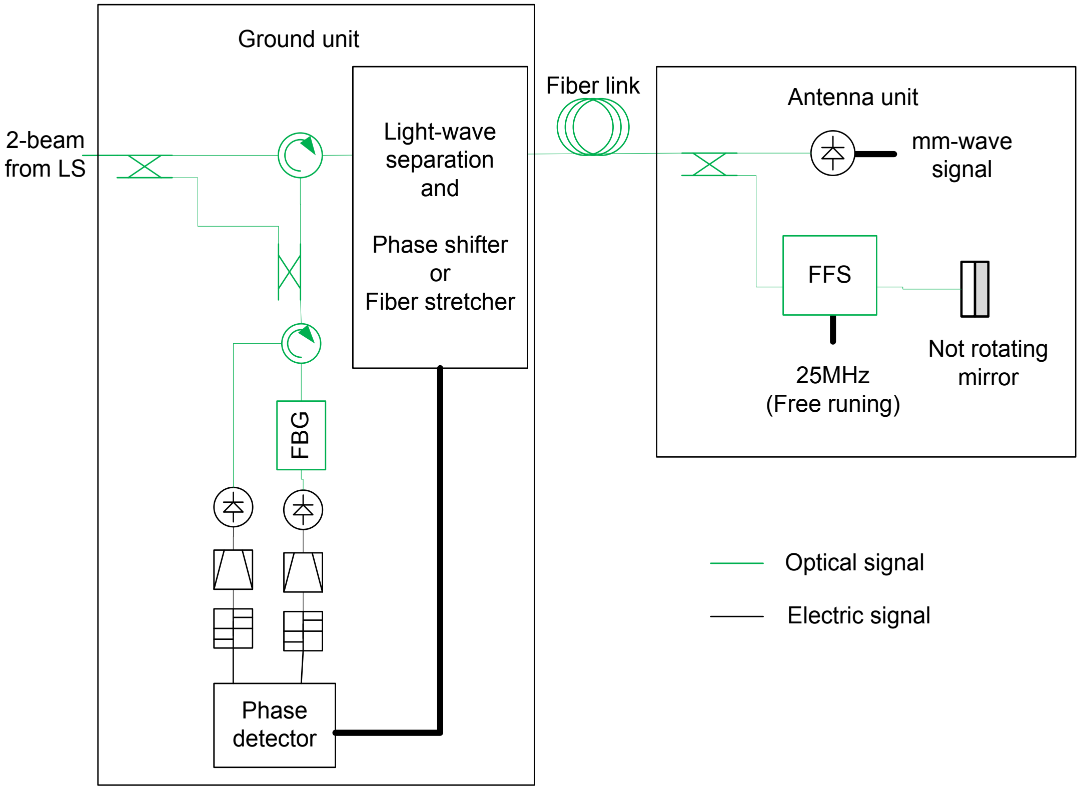


Fig. 7 Illustration of the Alternative Line Length Corrector scheme (ALLC), this method uses a fiber Bragg grating (FBG) to separate the master and slave laser, and then measure their phase change independently, to in this way measure the phase of the RF signal modulated on the optical carrier.

The ALLC scheme was also developed relying on the working hypothesis that it is sufficient to lock the fiber optical path length to obtain phase stability of the different frequencies needed. Moreover, it also assumes that it is not necessary to stabilize the fibers to the precision of the optical wavelength. The stabilization of the ALLC may therefore be relatively loose compared to the current LLC scheme and therefore less prone to fast perturbations. This approach has been tested in Japan with good results [29][30]. It was, furthermore, positively evaluated by NRAO but given the state of advance of the ALMA system design at the time of its proposal, it was not included into the final hardware implementation.

1.4.6 Temporal Drift Due to Fiber Movements

When an optical fiber is subject to external perturbations the SOP of the output will change. This change is denoted as the “State of Polarization Change” (SOPC). The SOPC is defined as the angular change between the SOP measured at the end the fiber before and after a perturbation is applied to the fiber. It can be calculated as

$$SOPC = \sin^{-1}(\vec{S_1} \cdot \vec{S_2}) \qquad \textbf{(1)}$$

with $\overrightarrow{S_1}$ and $\overrightarrow{S_2}$ the normalized non-intensity components of the Stokes vector that defines the SOP for both conditions.

Most of the SOPC comes from the fiber optics bending due to antenna movements. To reduce this problem a “fiber wrap” has been implemented by the engineering group of ALMA. In this device the fiber is bent without sharp angles and twists. With this, the DGD change produced by the fiber optics movement is small but still enough to change the SOP completely (in fact the SOPC can be random between 0 to π). On the other hand, as the fiber wrap is located at the end of the transmission system, the polarization change will not result in a variation of the input PMD vector, especially not in a change of the magnitude of the DGD for the whole fiber optical link. Given that the photo receiver also produces PMD, it interacts with the SOPC of the incoming signal producing a temporal drift of the output signal. It has been found that the maximum phase drift of the fiber output signal due to a small SOPC can be approximated by [31]:

$$0.55 \cdot DGD \cdot \sin(SOPC) \qquad \textbf{(2)}$$

1.5 Contributions of this work to the state of the art

During this work, a fast algorithm to optimize the input polarization to an optical fiber stretcher was developed. The objective of this optimization is to minimize the state of polarization change produced by the fiber stretcher when it moves. This algorithm was implemented and tested using the production fiber stretcher installed on the ALMA Line Length Corrector. This work lead to a publication in the IEEE Journal of Photonics [32] and in the SPIE international conference for astronomical instrumentation [33].

This paper also describes a method to have an accurate polarization control using a polarization rotator and a polarimeter separated by the LLC fiber stretcher. The algorithm does not make any assumption for the fiber stretcher optical transmission matrix. Therefore, it can be used in any situation where there exists an uncertainty in the transmission matrix between the polarization rotator and the polarimeter.

This can be useful to reduce the hardware costs of a system like this, eliminating the need for a second polarimeter, with a relatively small increase in optimization time, or it can be used as an update to an existing system. In the case of ALMA this algorithm allowed a significant reduction of the time required to perform the optimization of fiber stretcher input polarization with no hardware modification, and as in an observatory such as ALMA the observing time is never enough, this was an appreciated contribution.

Two PMD measurement campaigns where conducted, one at the Very Large Array (VLA) in New Mexico and another at the ALMA operations site at the Chajnantor plateau. These measurements allowed a better risk assessment of the PMD effect on the photonic central local oscillator system of ALMA.

1.6 Scope of this thesis

In chapter 2 this document explains a set of selected methods currently used to measure PMD in optical fibers. The report of the onsite PMD measurement campaigns performed at the VLA and ALMA observatories, together with laboratory measurements for characterization of the LLC fibers stretcher using a band 6 photomixer are reported in chapter 3. The conclusions from the results of each measurement set played an important role in steering the research of this thesis, by stimulating a change of direction and enabling the development of the new LLC calibration algorithm described in Chapter 4. Such contribution produced a significant reduction in execution time and accuracy of the polarization optimization algorithm used in the ALMA LLC. This new algorithm, based on Jones matrix Eigen analysis (JME), can potentially be applied in any field outside ALMA. Finally, the conclusions and perspectives for future work are presented in chapter 5.

2 Polarization Mode dispersion measurement techniques

2.1 The necessity for PMD measurement.

PMD is an undesired property of optical fibers that produces degradation of the optical signals they transmit, limiting the maximum speed of communication links. It also degrades the phase stability of the fiber link, limiting the ability of the fiber to transmit precisely high frequency timing signals [18].

The measurement of this property on installed fibers and in laboratory testing is a necessity for modern optical-fiber industry, for troubleshooting and for quality control. For this reason many methods to measure PMD exist. In the following lines we describe those which were used during the development of this thesis. The following sections are based on the original papers that describe the methods for the first time and also in the excellent review paper of Williams [34].

2.2 Previous definitions

2.2.1 Stokes vector and the Poincare sphere

The Stokes vector is a mathematical representation an electromagnetic wave, which describes its state of polarization. This vector has four elements known as Stokes parameters. They are usually named I, Q, U, and V, where I or S_0 is the intensity of the beam, and Q, U, and V or S_1, S_2, and S_3 represent the state of polarization of the beam. Table 1 shows few illustrative examples of stokes vectors therefore the Stokes vector can be Written as (3).

Table 1 : State of polarization represented by some interesting stokes vectors, the angle indicated for linear polarization, is measured from the positive direction of the x axis in a right hand coordinate system.

Horizontal linear(0°)	45° linear	Circular Right
$\begin{pmatrix}1\\1\\0\\0\end{pmatrix}$	$\begin{pmatrix}1\\0\\1\\0\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$
Vertical (90°) linear	-45° linear	Circular left
$\begin{pmatrix}1\\-1\\0\\0\end{pmatrix}$	$\begin{pmatrix}1\\0\\-1\\0\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\\-1\end{pmatrix}$

If Q, U, and V are normalized to be a unitary vector and then plotted on the Stokes space, they will be on the surface of the Poincare sphere; this is shown in Fig. 8. To plot them in this way is a practical way for visualizing the state of polarization of an optical signal. It is illustrative to notice that any linear polarization is situated on the equator of the sphere, being the positive direction of S_1 the horizontal polarization and the negative the vertical polarization. The positive direction of S_3 accounts for right handed circular polarization and the negative for left circular polarization.

$$\vec{S} = \begin{pmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{pmatrix} = \begin{pmatrix} I \\ Q \\ U \\ V \end{pmatrix} \qquad \textbf{(3)}$$

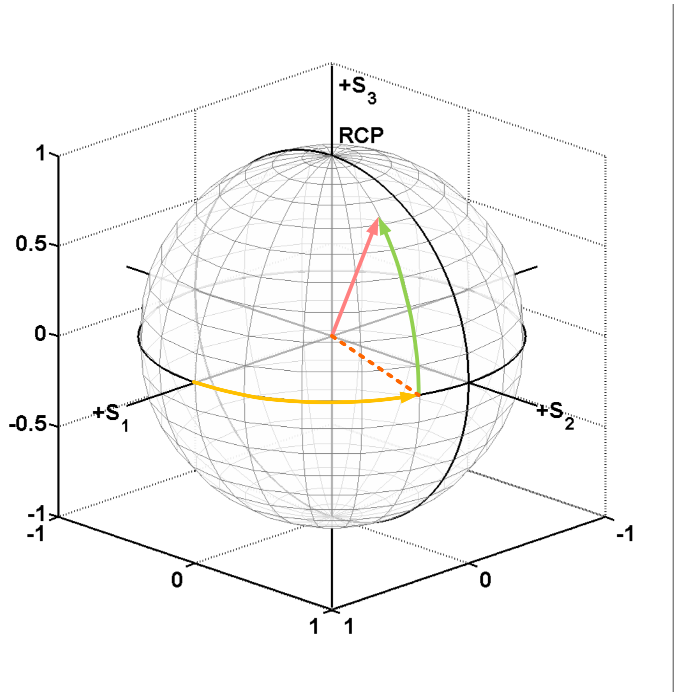


Fig. 8: Example of a Poincaré sphere in the Stokes space. A Stokes vector normalized to have an unitary magnitude appears as a point on the surface of the Poincaré sphere.

2.2.2 Jones Transmission matrix and Jones vector

The Jones vector is another representation of an electromagnetic wave. It uses the horizontal and vertical complex components of the electric field to describe the wave therefore can be expressed as in (4). Table 2 presents some illustrative examples of Jones vectors are presented.

$$\vec{J} = \begin{pmatrix} E_{0x} e^{i\emptyset_x} \\ E_{0y} e^{i\emptyset_y} \end{pmatrix} \qquad \textbf{(4)}$$

Table 2 : State of polarization represented by some illustrative Jones vectors. The angle indicated for linear polarization is measured from the positive direction of the x axis in a right hand coordinate system.

Horizontal linear(0°)	Circular Right
$\begin{pmatrix}1\\0\end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-i\end{pmatrix}$
Vertical (90°) linear	Circular left
$\begin{pmatrix}0\\1\end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix}1\\i\end{pmatrix}$

As the waves interact with the media in which they are traveling, the Jones vectors interact with the Jones matrix, which models the effect of the transmitting media on the waves. In Table 3 a few examples of Jones matrices for typical optical elements are presented.

Table 3: This table presents some examples of typical optical elements and their Jones matrix, in the last row an example of the mathematical description of the interaction between a wave and an optical element.

Description	Matrix representation
Vertical Linear polarizer	$\begin{pmatrix}1 & 0\\0 & 0\end{pmatrix}$
Quarter wave plate with vertical fast axis	$\begin{pmatrix}1 & 0\\0 & -1\end{pmatrix}$
Right circular polarizer	$\frac{1}{2}\begin{pmatrix}1 & i\\-i & 1\end{pmatrix}$
Interaction between horizontal polarized light, and a right circular polarizer	$\frac{1}{2}\begin{pmatrix}1 & i\\-i & 1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1\\-i\end{pmatrix}$

2.3 Jones-Matrix Eigen analysis to measure PMD

This method is one of the most popular because it produces a complete description of the PMD vector as function of wavelength for the measured optical component. The method requires the determination of the Jones matrix of the fiber or the optical component in at least two different wavelengths. To do this measurement the method proposed by Jones [35] was used as in Fig. 9. Performing this process, measurement for two different wavelengths produces the transmission matrixes $T(\lambda)$ and $T(\lambda + \Delta\lambda)$. Then, the PMD vectors are given by the Eigen vectors of $T(\lambda)T^{-1}T^{-1}(\lambda + \Delta\lambda)$ and the differential group delay (DGD) is given by the difference between the Eigen values of the same matrix [36].

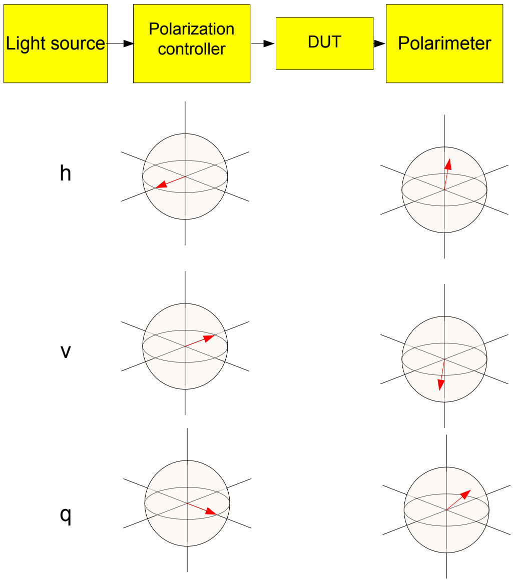


Fig. 9: Three input polarizations are sent into the device under test (DUT), vertical, horizontal, and another linear polarization signal located in the bisector of the first two. The output of each of the inputs is measured and the transmission matrix is computed using equations (5) and (6). The yellow boxes on top illustrate the hardware setup required to measure the transmission matrix, below the polarization controller the set input SOPs are shown and below the polarimeter the output SOPs are plotted.

$$T = \begin{bmatrix} k_1 k_4 & k_2 \\ k_4 & 1 \end{bmatrix} \qquad \textbf{(5)}$$

$$k_1 = \frac{h_x}{h_y}; k_2 = \frac{v_x}{v_y}; k_3 = \frac{q_x}{q_y}; k_1 = \frac{k_3 - k_2}{k_1 - k_3} \qquad \textbf{(6)}$$

where h_x, h_y, v_x, v_y, q_x, and q_y are complex Cartesian component of the Jones vectors measured as presented in Fig. 9 and T is the Jones transmission matrix for the DUT.

The working principle of this method is the rotation of the Stokes vector as the wavelength is swept. This rotation is produced by the birefringence of the transmitting medium. As the wavelength changes, the relative phase of the polarization component aligned with fast and slow axis of the medium changes, which effectively rotates the output polarization. The rotation axis for this movement over the Poincare sphere is the PMD vector. Moreover, the angular rotation rate is proportional to the vector magnitude which is the DGD of the medium at the wavelength used for the measurement.

The size of the wavelength step is a critical issue. As the step size increases the uncertainty of the measurement is reduced, but depending on the value of the DGD of the device under test (DUT) the measurement can become ambiguous if the SOP at the measurement point rotates more than $\pm\pi$ radians. The implication of this is shown in (7), where $\Delta\tau$ is the DGD in seconds and $\Delta\omega$ is the frequency step size in Hertz.

$$\Delta\tau\Delta\omega \leq \pi \tag{7}$$

If the DUT is a mode coupled device, like an optical fiber, the restriction is stricter. Simulations have shown that increasing the sample density increases the accuracy of the measurement. This is produced by the presence of second order PMD effect, (wave length dependency of both the DGD and PSP) then, to be sure that the measurement is within the 95% of the true DGD, and applying (7) to a center wavelength of 1550 nm we get (8) [34].

$$\Delta\tau\Delta\lambda \leq 1.5ps \cdot nm \tag{8}$$

With $\Delta\tau$ is DGD in ps and $\Delta\lambda$ is wavelength in nm.

2.4 Low Coherence Interferomentric Method.

This method is based on the straightforward idea of direct measurement of the time of flight of a signal traveling in the fast axis of the DUT, and compare it with signal traveling in its slow axis. However, instead of measuring the absolute time of flight, it measures the relative delay between them. To do this, a broad band light source is connected to one end of the DUT, and at the other end an optical interferometer is connected. As the movable arm scans a delay range, the central (delay ≈ 0) autocorrelation peak of the source will appear. If the DUT is a non-mode coupled device, two symmetrical peaks at each side of the central autocorrelation peak will appear additionally [37]. These two peaks correspond to interference between the fast axis light with the slow axis light, being the delay difference between these two peaks which is 2 times the DGD of the DUT, to achieve the interference of these orthogonally polarized signals, a quarter wave plate is inserted in one of the arms of the interferometer[38].

The width of the peaks is determined by the coherence time of the radiation source and the stability of the DGD across the light source bandwidth. If the coherence time increases or if the DGD has wavelength dependence, the peak can suffer some broadening.

If the DUT is a mode-coupled device, additional interference peaks will appear. The number of these peaks is a function of the number of mode coupled regions in the device. If the DUT has N mode coupled regions, the delay histogram will have $2^{(N+2)} - 1$ peaks, but many of these peaks will be merged, because the delay differences between many of these peaks will be smaller than the coherence time of the source. In consequence the final histogram will resemble a Gaussian curve. Finally, It was found that the standard deviation of the histogram, calculated as in (9), is approximately proportional to the RMS DGD of the DUT (10) [34]:

$$\sigma_M = \sqrt{\frac{\int I(t)t^2 dt}{\int I(t)dt}} \tag{9}$$

$$\langle \Delta\tau^2 \rangle^{1/2} \approx \sqrt{\frac{3}{4}}\sigma_M \tag{10}$$

Where $I(t)$ is the amplitude of the delay histogram and t is the time delay that produces this amplitude.

The equation (10) becomes exact in the limit of high PMD or a broadband source [39]. This approximation needs to be used with caution given that it assumes that the histogram is Gaussian shaped, but the autocorrelation peak and the measurement noise floor do not respond to this criterion, as show in Fig. 10.

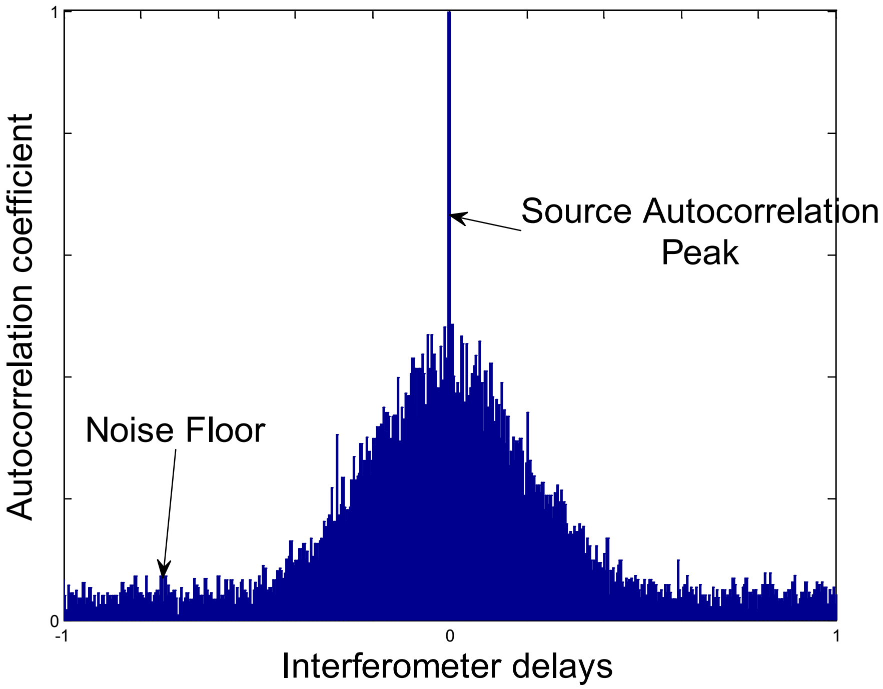


Fig. 10: Recreation of a delay histogram of a highly mode-coupled device, indicating the non Gaussian components; arbitrary units were used in this plot.

2.5 RF Phase-shift technique

This technique can be seen as another way to measure the time of flight difference between the fast and slow axis of the DUT. To perform the measurement, a tunable laser carrier is modulated by a radio frequency (RF) signal. Then it is sent across the DUT to a high speed photo detector where the signal is demodulated and its phase is compared with the phase of the original RF signal. When the light of the laser is transmitted by the fast axis of the DUT the phase difference will be minimum, and it will be maximum if the light is traveling by the slow axis. Then the DGD can be calculated as (11):

$$\Delta\tau = \frac{\delta}{360° \cdot f_M} \tag{11}$$

Where δ is the maximum phase change, measured in degrees, f_M is the frequency of the modulated signal in Hertz, and $\Delta\tau$ is the measured DGD in seconds. To use (11), the polarization state of the optical signal needs to be aligned to the fast or slow axis (PSP) of the DUT, but this is not a trivial task. One approach can be to scan the Poincare sphere and measure the RF phase at each polarization state [40], but this is time consuming and increases the chance of phase drifts due to environmental disturbances across the optical path of the signal. This effect can cause significant over estimation of the DGD.

A number of approaches have been proposed to speed up the measurement. The most cited one uses only four launched polarization states to determine the PSPs and the DGD [28], the selection of these polarization states is simple, three of them must be within a great circle of the Poincare sphere and separated by 90 degrees, and the fourth one must be on the axis of this circle.

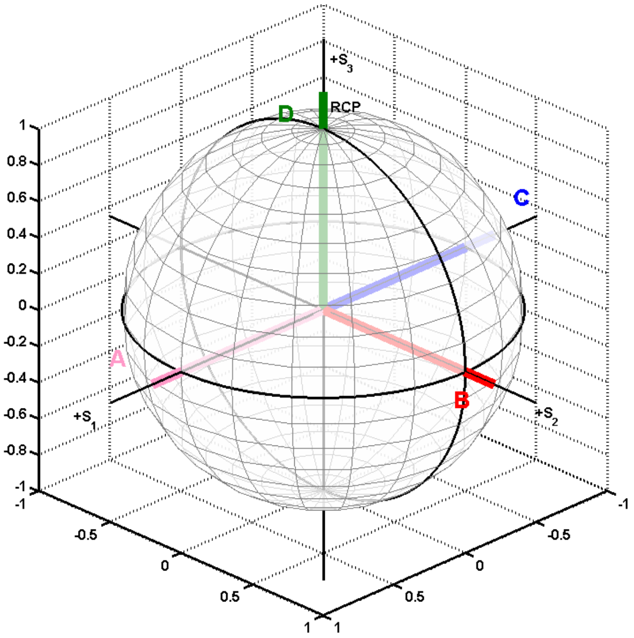


Fig. 11: Poincare sphere with the four launch polarizations used for the RF phase shift method to measure PMD

After choosing the polarization states as in Fig. 11, δ can be calculated as

$$\delta = 2\tan^{-1}\left[\left(\tan^2(\phi_{RF,A} - \Phi) + \tan^2(\phi_{RF,B} - \Phi) + \tan^2(\phi_{RF,D} - \Phi)\ \right)^{\frac{1}{2}}\right] \quad \textbf{(12)}$$

with

$$\Phi = \frac{(\phi_{RF,A} + \phi_{RF,C})}{2} \quad \textbf{(13)}$$

and $\phi_{RF,A}$, $\phi_{RF,B}$, $\phi_{RF,C}$, $\phi_{RF,D}$, the measured phase change for the polarization states A, B, C, D. Then, using (11), the DGD can be calculated [28]. The two PSP can also be calculated using the equations described in [28].

This method is capable of performing true narrowband PMD measurements having a spectral resolution that can be even better than the Jones Matrix Eigen analysis (JME). This method is well suited to measure PMD in optical devices that do not produce optical signals at their output as it occurs with photomixers.

3 Preliminary PMD Measurements

3.1 Extended Very Large Array measurement Campaign

3.1.1 Overview

The PMD is an important parameter of an optical fiber link, especially if the highest performance is pursued. In the case of ALMA's central local oscillator (CLO), two aspects of the PMD can be an important limiting factor for the phase stability. They are its magnitude and its rate of change. In this chapter the results of a short measurement campaign are shown. These measurements were made over two different optical fibers at the Extended Very Large Array (EVLA), New Mexico. They had a length of 44 and 14 km, respectively.

The EVLA is radio interferometric array similar to ALMA, it consist of 27 25 m parabolic antennas, arranged in "Y" shaped layout, with baselines of up to 36 km. Its operating frequency range is 1.4 GHz to 43.5 GHz. Originally built in the seventies, it was denominated Very Large Array (VLA) but recently a complete update of the original system was made and the word "Extended" was added to the name to account for the enhanced capabilities that the new array has.

3.1.2 Objectives

The objectives of the reported measurements at the EVLA were to evaluate two different instruments to make PMD measurements, and to have a first view of the behavior of the PMD in buried fibers like those to be installed also at ALMA.

3.1.3 Methodology

3.1.3.1 Instruments

The measurements were made with two different instruments. The first one was the "EXFO FTB-5500B" which uses the interferometric method to measure the average DGD over the C[7] and L[8] optical bands. It requires a broad band optical source connected to the far end of the fiber. The optical source used in the set up was the EXFO FLS5800 that covers both bands.

The other instrument was the Agilent "N7788B Bench top Optical Component Analyzer" which uses the JME method to measure the full PMD vector as function of the wavelength. It requires the use of a tunable laser connected directly to the instrument. The input and output of the DUT must be also connected to the instrument. To make this connection possible, a loop was made using two optical fibers buried in parallel, in this way the signal emitted by the instrument makes a round trip by the buried fiber and returns to the lab.

[7] Optical band C covers: 1530-1565nm

[8] Optical band L: 1565-1625nm

3.1.4 Setup

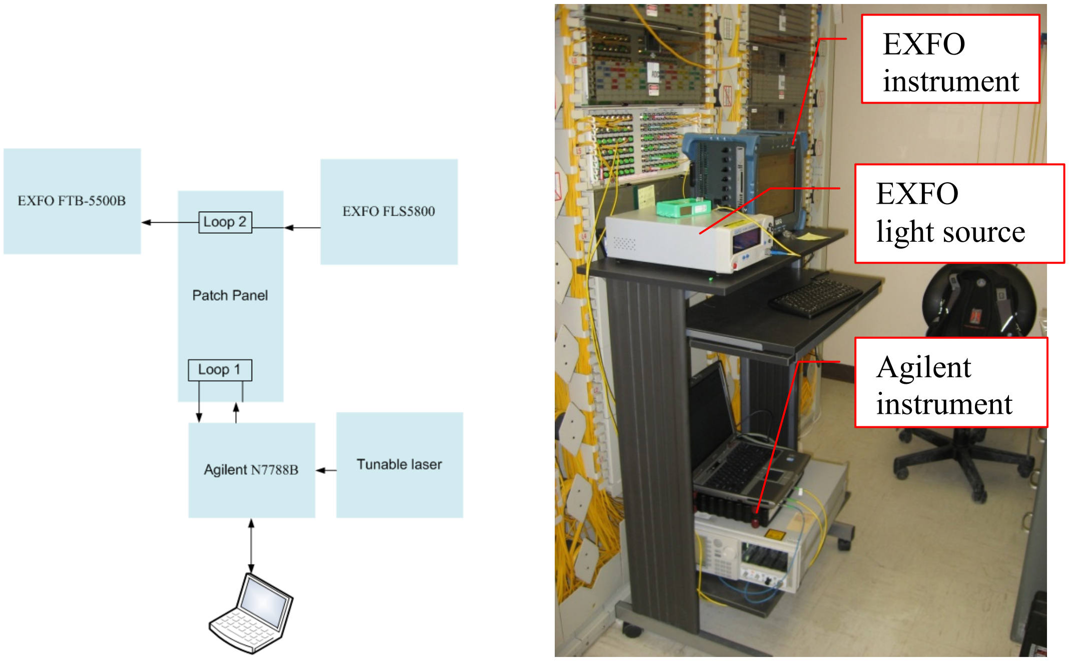


Fig. 12 Left panel: the block diagram of the setup. In the case of the EXFO instrument the light source is connected to one end of the fiber loop to feed the required broad band optical signal, and the instrument itself is connected to the other end of the fiber loop. Right panel a picture of the actual arrangement in the patch panel room.

In Fig. 12 the actual setup used to perform the measurements is presented. To make more efficient use of the available time, both instruments were used at the same time connected to different fiber loops. These loops were made at the antenna pad AE9 with a round trip length of 44 km and the BE9 with a round trip length of 14 km. The pads configuration can be found in the diagram of Fig. 13.

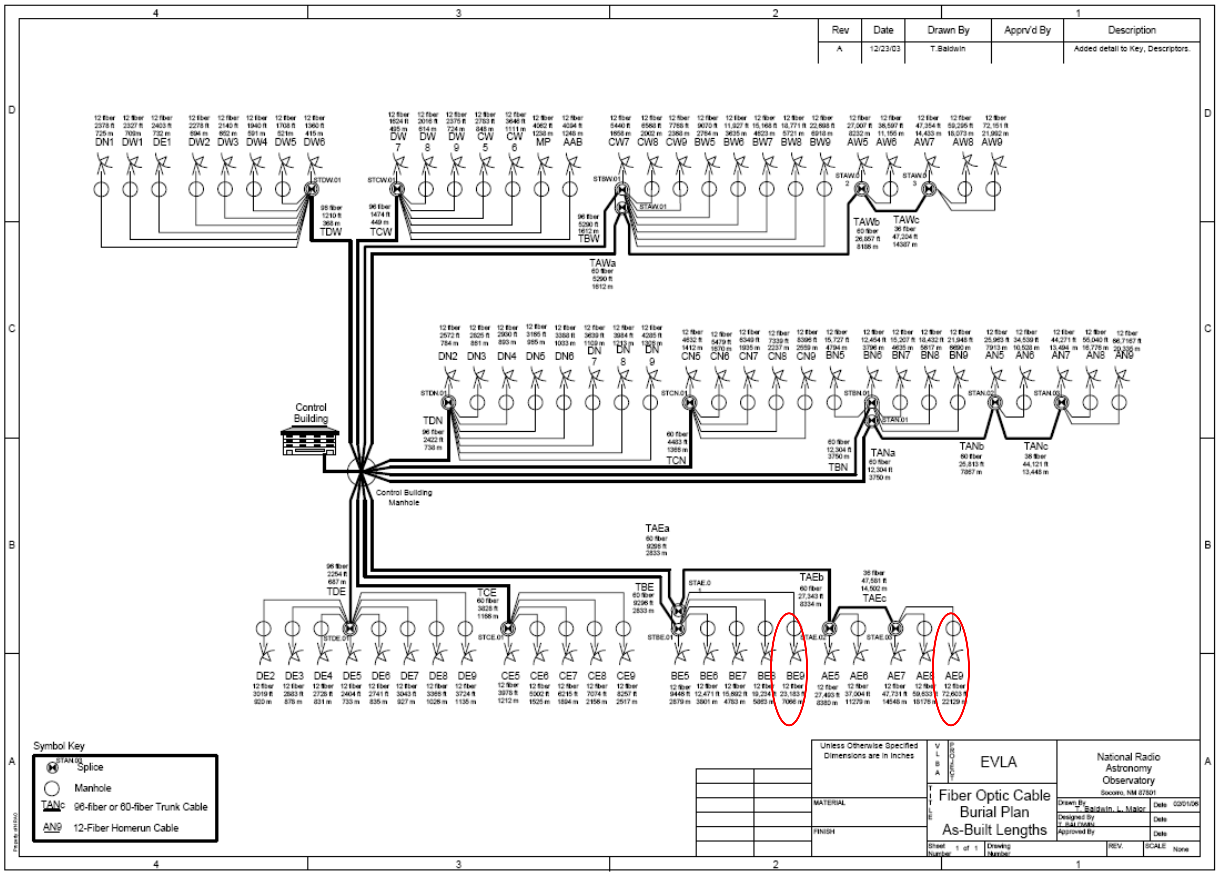


Fig. 13 Fiber plan of the EVLA. Pads used for the measurement are circled in red.

3.1.5 Results

3.1.5.1 Agilent N7788B (Jones matrix Eigenanalysis (JME) method)

The behavior of the instrument was not completely satisfactory, the software still had many bugs and its usability could be improved. For example, internal data analysis requires a long time and, in some cases, data was lost due to a bad software configuration. On the other hand, the instrument gives a wealth of information beside the DGD, for example, the PSP and DGD as function of wavelength and time. There is enough flexibility in how to retrieve the data to external programs, e.g. Matlab, which was subsequently used here to analyze the data thoroughly.

3.1.5.1.1 PMD and SOP stability on a 44 Km buried fiber AE9 at the EVLA site, measured with the Agilent N7788 optical component analyzer

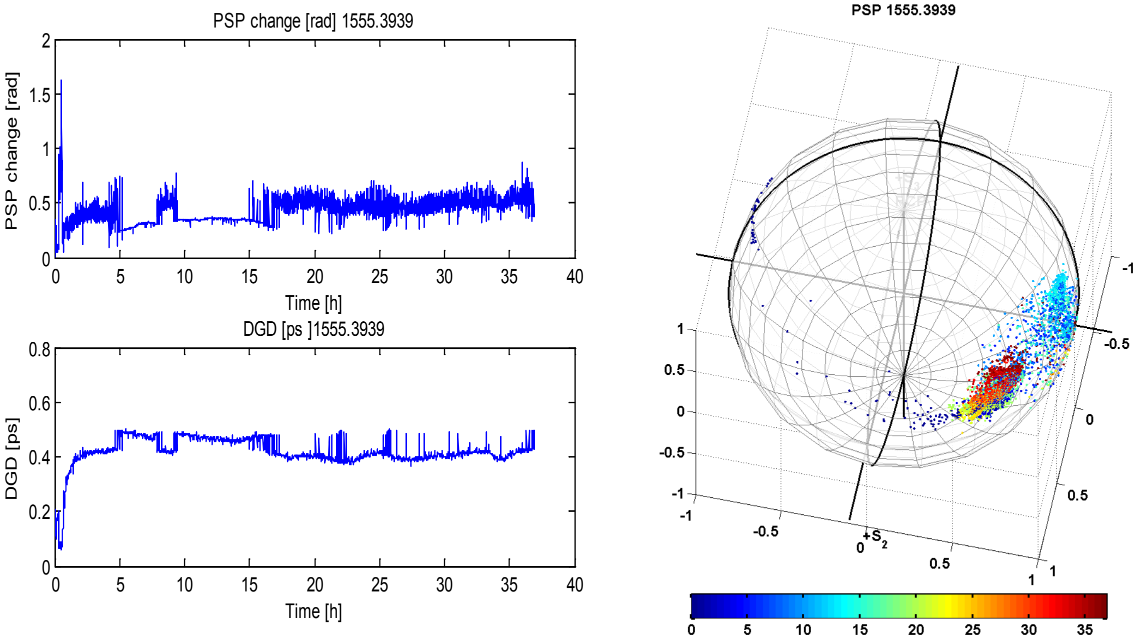


Fig. 14 The plot on the top left shows the change in the direction of the PMD vector as function of time. The plot in the bottom left shows the value of the DGD. The sphere on the right shows the position of the PMD vector over the Poincare sphere during the measurement. These measurements were performed at 1555.39 nm, the color coding shows the elapsed time in hours.

Fig. 14 shows the PMD vector as function of time for a specific wavelength. It is clear that the change in the direction of the PSP during the measurement was big, with a value of approximately $\pi/2$ over the Poincare sphere if the initial transient is neglected. The DGD shows a strong change during the first 2 hours of the measurement, probably originated by the presence of people in the patch panel room during this time. It is notable that this measurement shows a two states behavior. This is clear in the DGD plot between the hours 5 and 28, showing like to lines plotted alternately. This behavior can be seen also in Fig. 15 to Fig. 18. In order to understand it the vendor was contacted but his explanations where not satisfactory, indeed they did not have a clear idea of why this happened.

The measurement of the SOPC was made by retrieving the transmission matrices recorded by the instrument into Matlab. To make the plots shown in Fig. 17 and Fig. 18, these matrices were used to calculate the output polarization for a hypothetical horizontal input polarization. In this case the two states behavior is also noticeable.

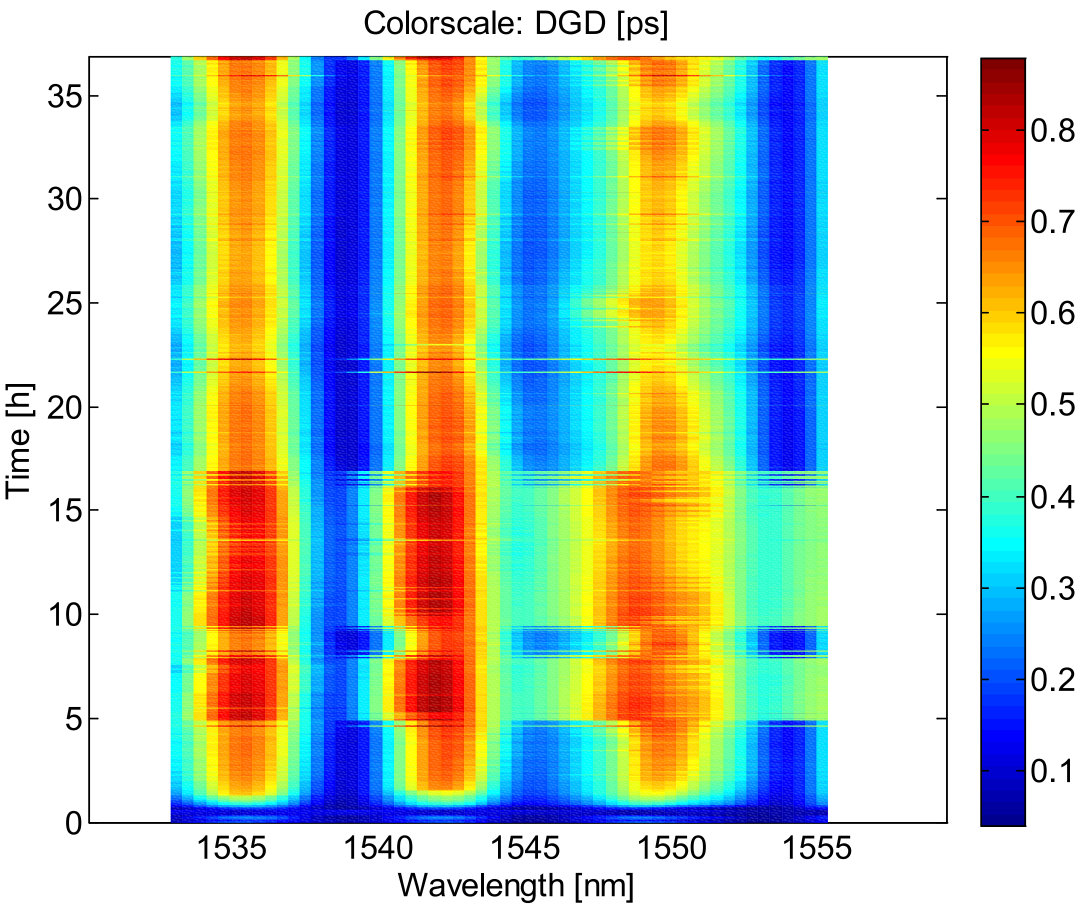


Fig. 15: DGD as function of the wavelength and time for the 44 km of buried fiber. Despite that the bi-stable behavior is also present in this plot and is a pure function of time at every wavelength, it can be noticed a strong wavelength dependence of the DGD. Beside the bi-stable behavior, there is not much time dependence of the DGD visible in this case.

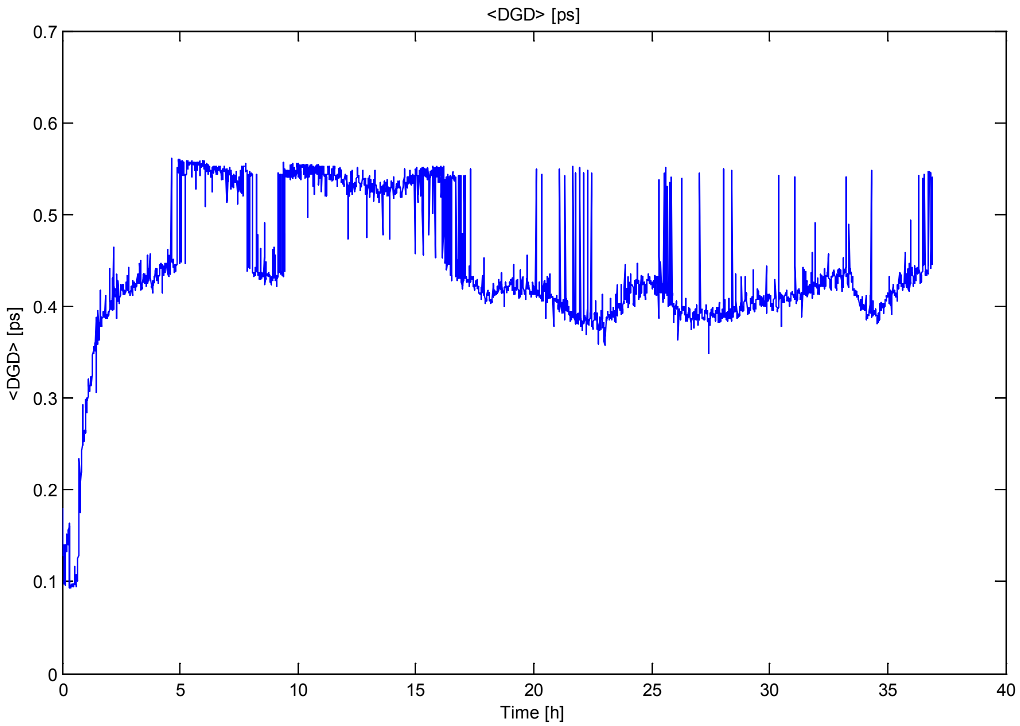


Fig. 16 Average DGD over 1530 to 1560nm as function of time

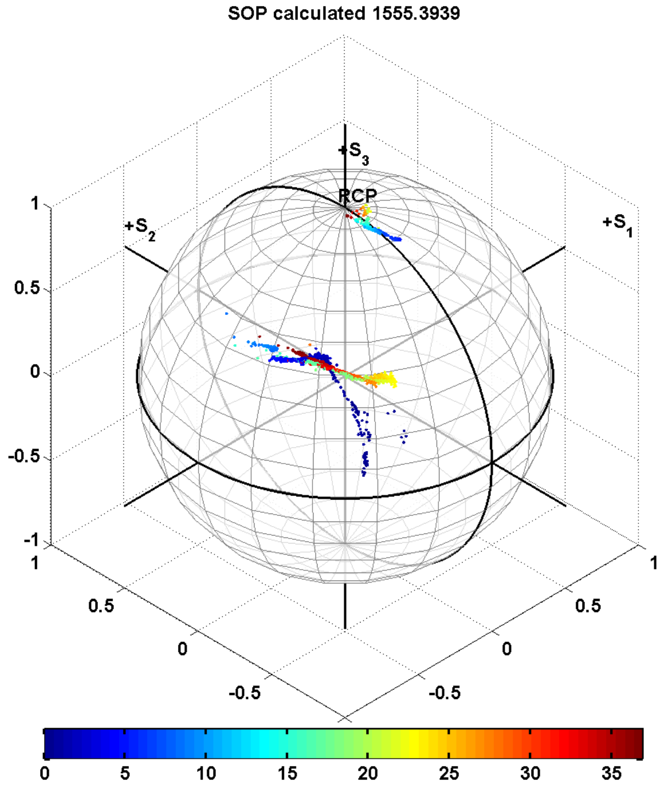


Fig. 17. SOP calculated with the Jones matrices stored by the instrument. The color code is in hours.

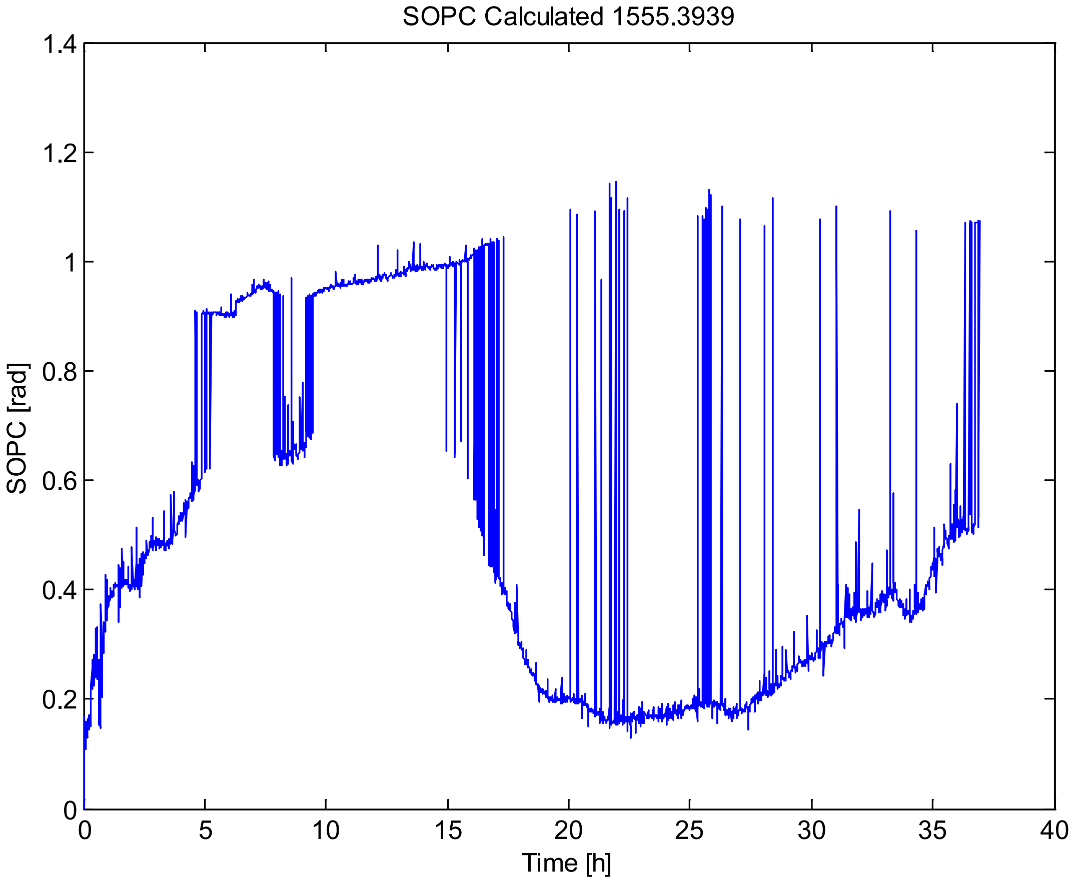
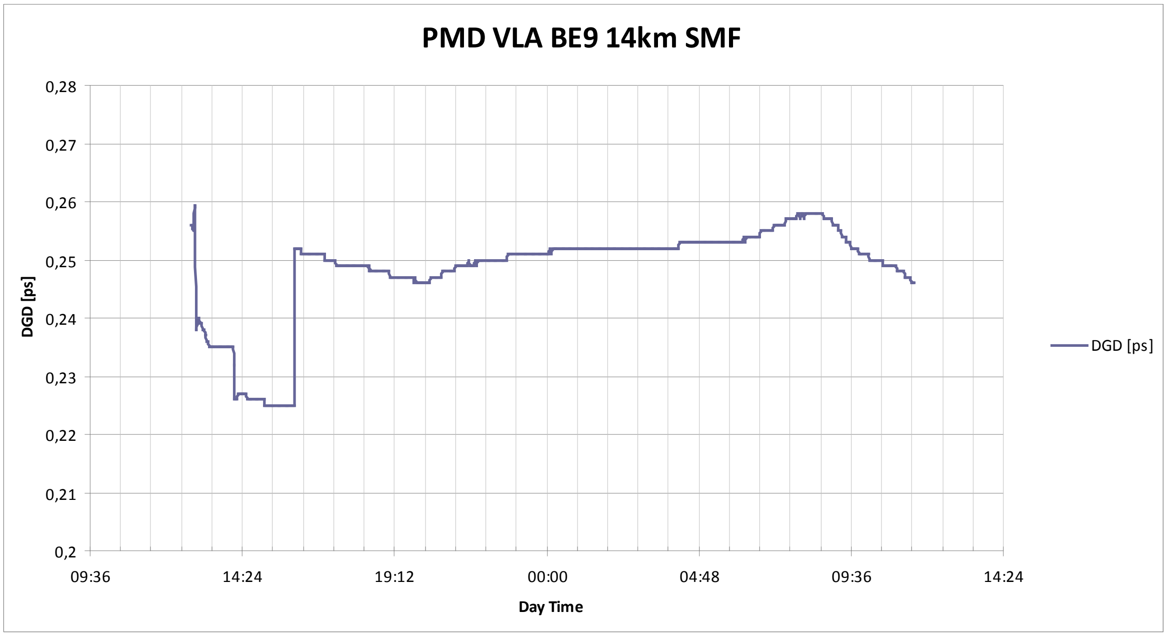


Fig. 18: SOPC of the 44km fiber, calculated from the Jones matrix stored by the instrument.

3.1.5.2 *Interferometric method*

3.1.5.2.1 DGD of buried fibers at the VLA site, measured with the EXFO FTB-5500-B PMD analyzer



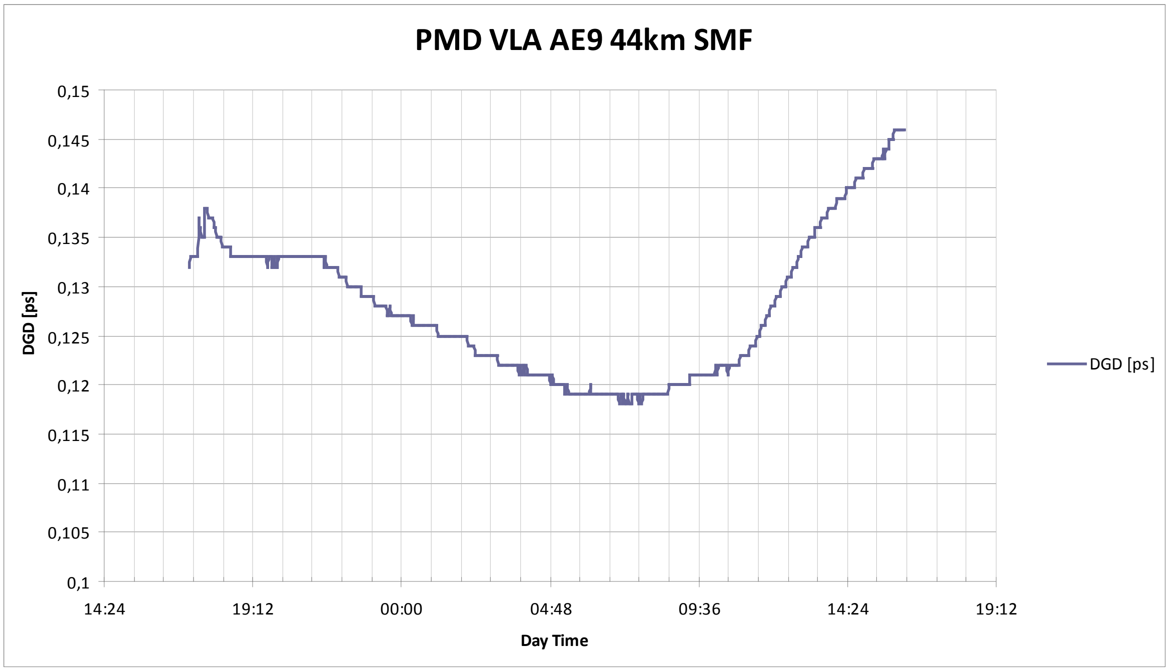


Fig. 19 Average DGD measured by the EXFO instrument over several hours. (a) DGD for the 14-km fiber. (b) DGD of the 44-km fiber.

The data taken with this instrument only shows the average DGD over the optical bands C and L. However, it has the advantage that the instrument is very easy to use and after completing the physical connections, a single button does the entire job. This instrument does not allow an easy access to the data, because it has a closed file format that cannot be retrieved directly to other software like Matlab. To retrieve the data, a non-straight-forward procedure was performed by porting the data to html format and opening the file with MS Excel.

Fig. 19 shows the behavior of the DGD for the 14 km and the 44 km long fibers at the VLA. The 14 km fiber was somehow perturbed during the data collection, making the first two hours of data non-valid. After this period of time the data was smoothed and showed a maximum variation of 4%, this, in contrast to the variation of the 44km-fiber, which was around 20%. This fiber presented a bigger relative and absolute variation. In spite of that, the DGD was smaller in the 44 km fiber than in the 14 km fiber.

3.1.6 Summary

A PMD measurement campaign with two different instruments over two different fibers was made with the purpose of evaluating the performance of two complementary instruments:

1.) The Agilent instrument confirms that it can produce very rich information about the state of the link, having an excellent flexibility. As a consequence of all this flexibility, the use is complicated with too many variables to set, making easy to make mistakes that can produce corrupted measurements. Moreover, the software is still buggy and its usability is not one of its best features, making the data analysis a time consuming task. The data that the instrument produces through the output link is very rich, it includes the PSP and DGD, as function of wavelength and time, and it contains also all the raw data obtained from the instrument. This, added to the possibility of retrieving the data to Matlab, makes it a really interesting research instrument.

2.) The EXFO instrument has a completely different approach. It is very easy to use and, as a consequence, it produces reliable information. However, a major drawback is that it only retrieves information about the average DGD over the bands C and L, and does not have the possibility of making further analysis of the data, which is kept inside the proprietary file structure of the instrument, not accessible by the user.

The data obtained using the Agilent instrument shows instability in the PSP of the 44 km link and an extreme change of the DGD of the link during the first two hours. The SOPC of the 44-km fiber is not small, especially within the first two hours of measurement. Unfortunately, the data taken for the 14-km fiber does not look good. It is extremely noisy and really not usable, and therefore was not included into this paper.

In conclusion, despite the problems found in the data provided by the Agilent instrument which uses the JME method to measure PMD, it can be stated that the information provided by JME is the best, making JME the most useful method for the purposes of this thesis.

3.2 PMD measurement campaign at the ALMA Array operations site (AOS)

3.2.1 Objective

The objective of this campaign was to have a first impression of the PMD performance, of the fibre link of the first bunch of antenna pads installed on the ALMA operations site (AOS). This was requested to evaluate the fibre optics contractor's work and to decide if the risk of PMD is still important for the success of the ALMA observatory.

3.2.2 Setup

During this campaign a set of 22 pads located approximately 500 m away from the AOS technical building, at 5000m above sea level[9], was used. Each of these pads has a fibre link that consists of a fibre cable with 8 fibres: one fiber for the photonics LO transmission, another to get the payload data from the antenna and two more used to establish an Ethernet communication link with the antenna, plus 4 spare fibres. All of them were measured using the PMD5000 instrument from Thorlabs. It consists of a deterministic polarization controller DPC5500 and a IPM5300 polarimeter, both made by Thorlabs, and an external tuneable laser also from Thorlabs. This instrument is based on the JME method, and is capable of producing high wavelength resolution PMD measurements, in a similar way as the Agilent instrument, as reported in the previous section. To perform the actual measurements, all this equipment was driven by a special purpose software provided by the same vendor.

Given that the length of the links available at the time of the campaign (in 2010) was relatively short (just the Morita array pads, which are about 500 m away from the central building), the actual measurements were made looping all 8 fibres of a cable together ("daisy chain"), increasing in this way the length to 4 to 5 km for each pad. This was necessary especially for the EXFO instrument where the PMD values were below the precision of the instrument, and was increasing the measurement precision for the Thorlabs instrument.

This measurement campaign extended for over a week at the AOS, sharing the site with heavy machines finishing the access roads to the used antenna pads. Therefore perturbed measurements had to be sorted out.

[9] These pads are known the Atacama Compact array and more recently renamed as Morita Array, honouring the Japanese astronomer Kohichiro Morita.

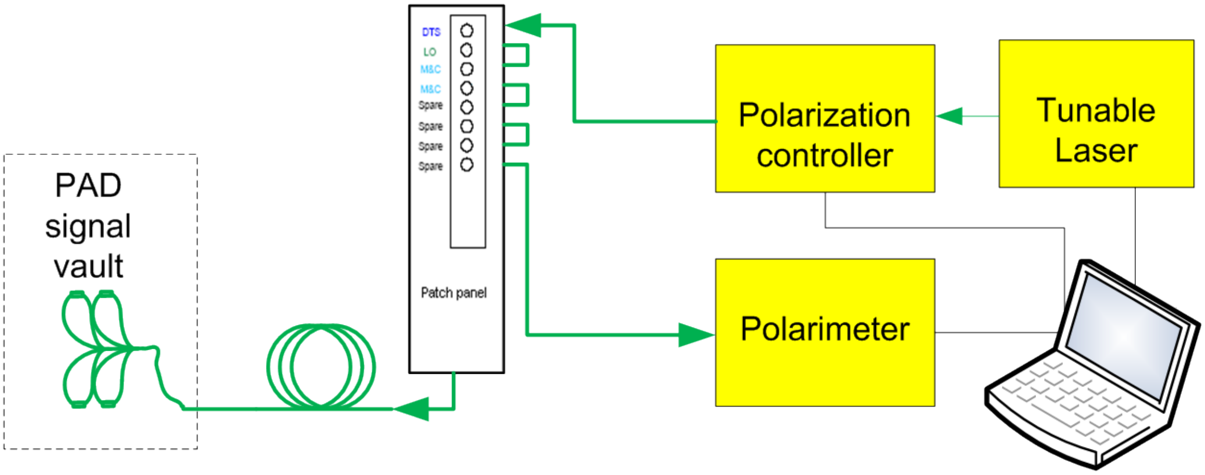


Fig. 20: This layout shows the hardware setup used to perform the measurements. This includes a tunable laser, a deterministic polarization controller and a polarimeter from Thorlabs, everything controlled from a laptop computer running the Thorlabs TPX PMD5000 software for PMD measurements.

3.2.3 Results

The plots Fig. 21 to Fig. 23, show the results of 3 days measurement logging over the pad J502 which 8-fold fiber length is 5.2 km. The SOPC plots and Poincare sphere (Fig. 21) were calculated using the transmission matrixes calculated by the instrument, assuming horizontal input polarization, the data shown in these plots represents the behavior of the result of this calculation.

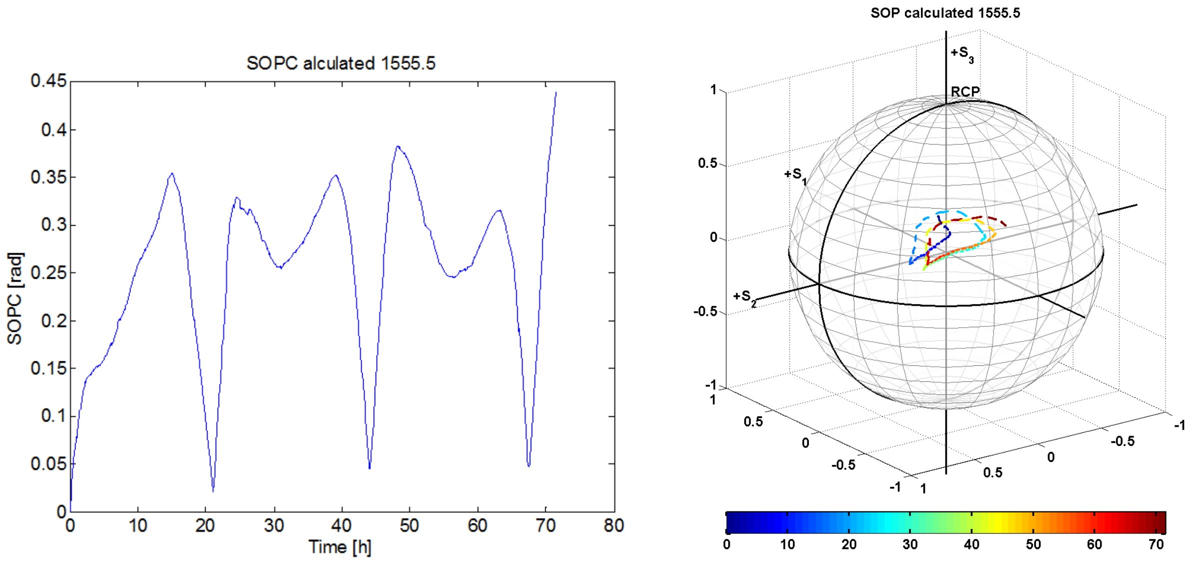


Fig. 21: The left plot shows the SOPC calculated from the Jones matrix stored by the instrument, and

the Poincare sphere at the right shows the evolution of the SOP. The color coding shows the elapsed time in hours. It can be seen a significant drift of the SOP during the measured time interval, which can be quite fast, around 0.07 radians per hour during day time. Despite this, the impact on the phase stability of the LO signal stays within the specified range.

The stability of the PMD vector, represented by variations of the DGD value and PSP orientation, is shown in Fig. 22. In this case the information was calculated directly by the instrument. From the plots it can be seen that the DGD is highly correlated with the time of the day when data was taken, with quite good repeatability. The direction of this change (PSP change), however, shows a more random behavior, being less repeatable than the DGD value. This gives a good explanation for the behavior observed for the SOP of the transmitted signal shown in Fig. 21.

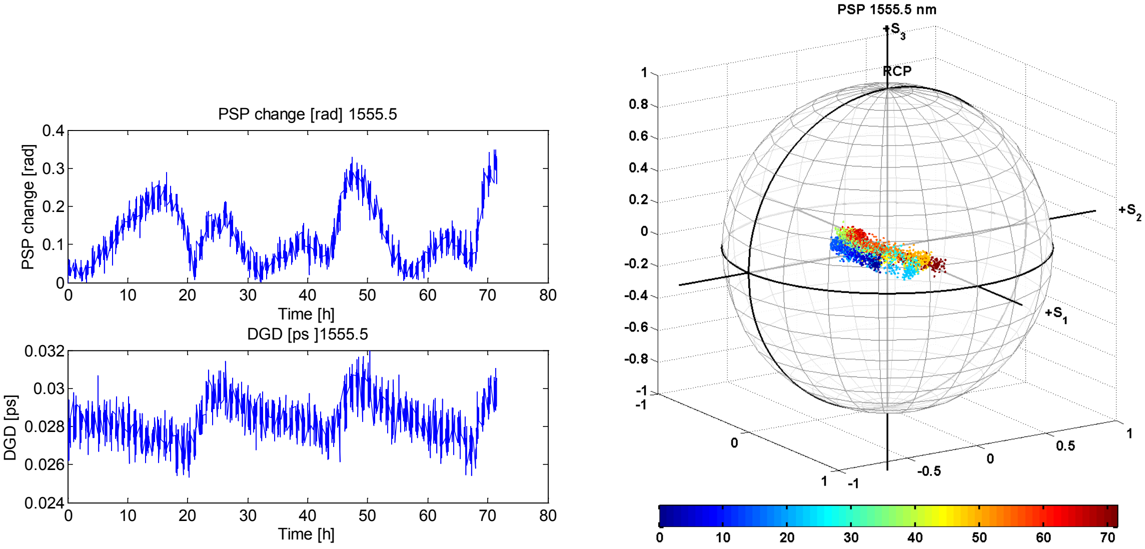


Fig. 22: The plot on the top left shows the change in the direction of the PMD vector as function of time, and the plot in the bottom left shows the corresponding value of the DGD. The sphere on the right shows the position of the PMD vector over the Poincare sphere during the measurement, which was made at 1555.5nm. The color coding shows the time variable in hours.

The plot in Fig. 23 shows the behavior of the DGD versus wavelength and time. It shows a periodic behavior in time as it appears already in Fig. 22. In this plot it can be seen also that for this wavelength range the DGD has periodicity as function of the wavelength, but the amplitude increases with wavelength showing deeper minima and higher maxima in the range 1580-1620 nm.

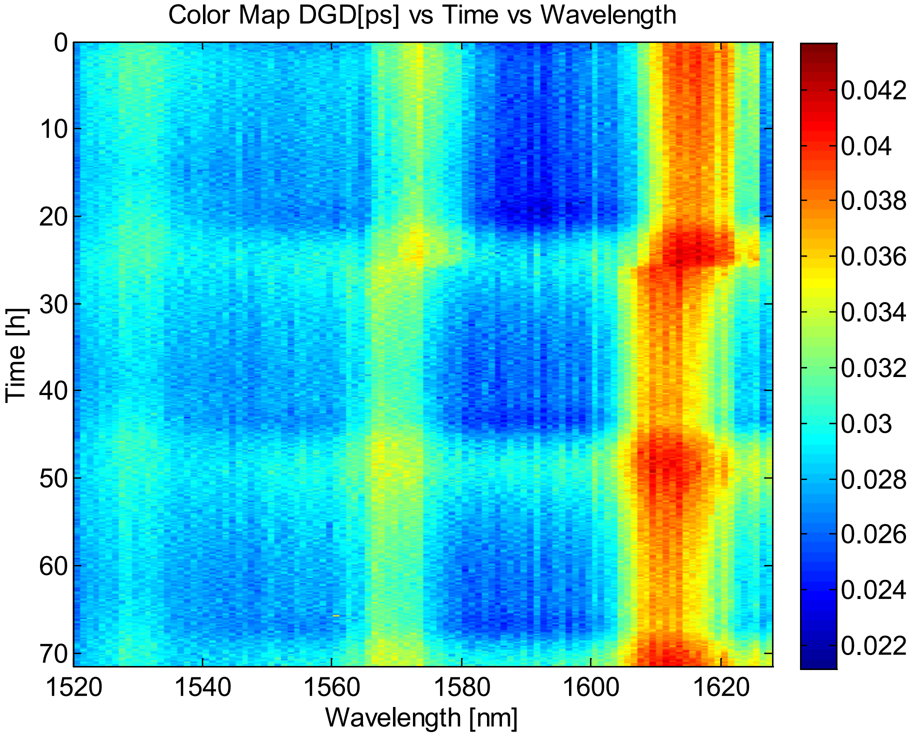


Fig. 23: This plot shows the DGD as function of the wavelength and time for the tested fiber. It shows a clear time dependence with a 24 hour period.

A summary of the results of 21[10] pads obtained at 1556 nm is shown in Table 4. This results correspond to a fiber loop made with the maximum available number of fibers for each pad (the 8 of the cable). It can be seen that the DGD coefficients of the fibers are in the order of a few tens of femtoseconds, which is significantly less than the ALMA specification of 0.05 $ps/\sqrt{km}$ [41] for the CLO fibers, and in no case the coefficient is bigger than that value.

[10] The J505 pad is not present in this table because the data for this pad was corrupted. Therefore only 21 of the 22 pads on the ACA pads are summarized in Table 4.

Table 4 : Summary of the results obtained @ 1556nm wavelength for a fiber loop made with maximum available fibers on each pad. The first column presents the pad ID, the next 3 columns present the different measurements made for each pad (each measurement was repeated 3 times, in order to reduce the instrumental errors by averaging) then the next two columns show the measurement average and standard deviation, the next three columns show the link length, the number of fibers looped to form the link and the last column presents the average DGD coefficient of the link in $ps/\sqrt{km}$.

PAD ID	DGD 1 [ps]	DGD 2 [ps]	DGD 3 [ps]	average DGD [ps]	STD deviation DGD [ps]	Link length [m] (OTDR measurement)	number of fibers in the link	average DGD coefficient $[ps/\sqrt{km}]$
J501	4.14E-02	3.95E-02	4.43E-02	4.17E-02	2.40E-03	4950	8	1.88E-02
J502	3.23E-02	3.25E-02	3.25E-02	3.24E-02	1.15E-04	5243	8	1.42E-02
J503	4.02E-02	3.95E-02	4.21E-02	4.06E-02	1.35E-03	5190	8	1.78E-02
J504	5.15E-02	5.07E-02	5.38E-02	5.20E-02	1.58E-03	5050	8	2.31E-02
J506	3.75E-02	4.17E-02	3.88E-02	3.93E-02	2.12E-03	5305	8	1.75E-02
J507	2.27E-02	2.26E-02	2.17E-02	2.23E-02	5.57E-04	5026	8	9.95E-03
J508	3.39E-02	3.82E-02	3.89E-02	3.70E-02	2.69E-03	4852	8	1.68E-02
J509	4.86E-02	4.70E-02	4.87E-02	4.81E-02	9.44E-04	3764	8	2.48E-02
J510	3.61E-02	3.60E-02	3.69E-02	3.63E-02	4.83E-04	5309	8	1.58E-02
J511	8.43E-03	1.81E-03	2.60E-03	4.28E-03	3.62E-03	2778	4	2.57E-03
J512	5.48E-02	5.31E-02	5.20E-02	5.33E-02	1.39E-03	2815	4	3.18E-02
N601	1.93E-02	1.82E-02	1.94E-02	1.90E-02	6.69E-04	4966	8	8.51E-03
N602	1.98E-02	2.32E-02	2.55E-02	2.28E-02	2.84E-03	4900	8	1.03E-02
N603	2.74E-02	3.30E-02	3.18E-02	3.07E-02	2.97E-03	4860	8	1.39E-02
N604	4.39E-02	4.33E-02	3.87E-02	4.20E-02	2.84E-03	5515	8	1.79E-02
N605	1.48E-02	9.68E-03	1.09E-02	1.18E-02	2.67E-03	5429	8	5.06E-03
N606	4.09E-02	4.17E-02	3.97E-02	4.08E-02	1.02E-03	5560	8	1.73E-02
T701	2.30E-02	2.26E-02	2.00E-02	2.19E-02	1.60E-03	4027	8	1.09E-02
T702	4.12E-02	3.94E-02	4.12E-02	4.06E-02	1.00E-03	5655	8	1.71E-02
T703	4.36E-02	4.46E-02	4.66E-02	4.49E-02	1.55E-03	5702	8	1.88E-02
T704	5.23E-02	5.13E-02	5.04E-02	5.13E-02	9.76E-04	4831	8	2.34E-02

3.2.4 Discussion

Form the data captured in this campaign it can by concluded that:

1. The PMD of the installed fibres of the first 21 pads exists, but is extremely low, well below the ALMA specification.
2. Despite the precaution invested to thermally insulate the fibres from the environment, the PMD shows a high correlation with the day time, but the DGD value was never beyond the specification.
3. As a consequence, the polarization stability of the optical signal reaching the antennas is stable enough to be sure that the phase stability of the PLO signal reaching the antennas will be well within the specified range.

3.3 Photomixer PMD measurement using phase-shift technique

3.3.1 Objective

This measurement was quickly setup to get a preliminary approximation to the PMD of the photomixers used in the ALMA warm cartridge assembly (WCA). This value was initially assumed to be very low, but the measurement showed something different.

3.3.2 Setup

The setup used for this measurement, as shown in Fig. 24, was made reusing the ALMA LLC/SAS test set. The original purpose of this test set was to test the phase stability of LO signal transmitted by the CLO after traveling across a 15 km fiber spool. For this test the fiber spool was removed and a deterministic polarization controller was added to the line length corrected link. With this, the behavior of the phase of the photomixer can be studied as a function of the input polarization. Then, using the technique explained in the section 2.5, the phase of the signal detected by the photomixer was monitored while the SOP of the optical signal feed to it was swept across the four SOPs shown in Fig. 11. The DGD of the photomixer was estimated, for a single photo mixer.

The polarization controller setting was manually moved to produce the required launch polarization states, while the phase measurement was logged. During the measurement the LS was locked at 80 GHz.

3.3.3 Results

Table 5: DGD Calculation for the photomixer, A, B, C, and D are the phase measurements required for the formula (12), and (13). Two calculations are shown one using just the raw phase data and other using just the magnitude of the phase jump when the launch polarization is changed, this in an intent of suppressing the phase drift experimented by the test set.

	A[°]	B[°]	C[°]	D[°]	Φ[°] (13)	δ[°] (12)	**DGD[ps]** (11)	Estimated Error [ps]
Drift suppressed	-1.08	1.54	-0.66	-0.01	-0.87	0.09	**0.18**	0.08
raw phase	-40.82	-38.76	-40.97	-39.90	-40.90	0.08	**0.16**	0.06

Table 5 presents the data and the calculation used to estimate the DGD of the photomixer. There is one calculation for the raw phase data and another using only the magnitude of the phase jumps observed when the launch polarization is changed. This is an intent to reduce the effect of the phase drift observed on the test set. In Fig. 25 the phase data logged during the measurement is plotted and is superimposed with the curve obtained using only the phase jumps as described before.

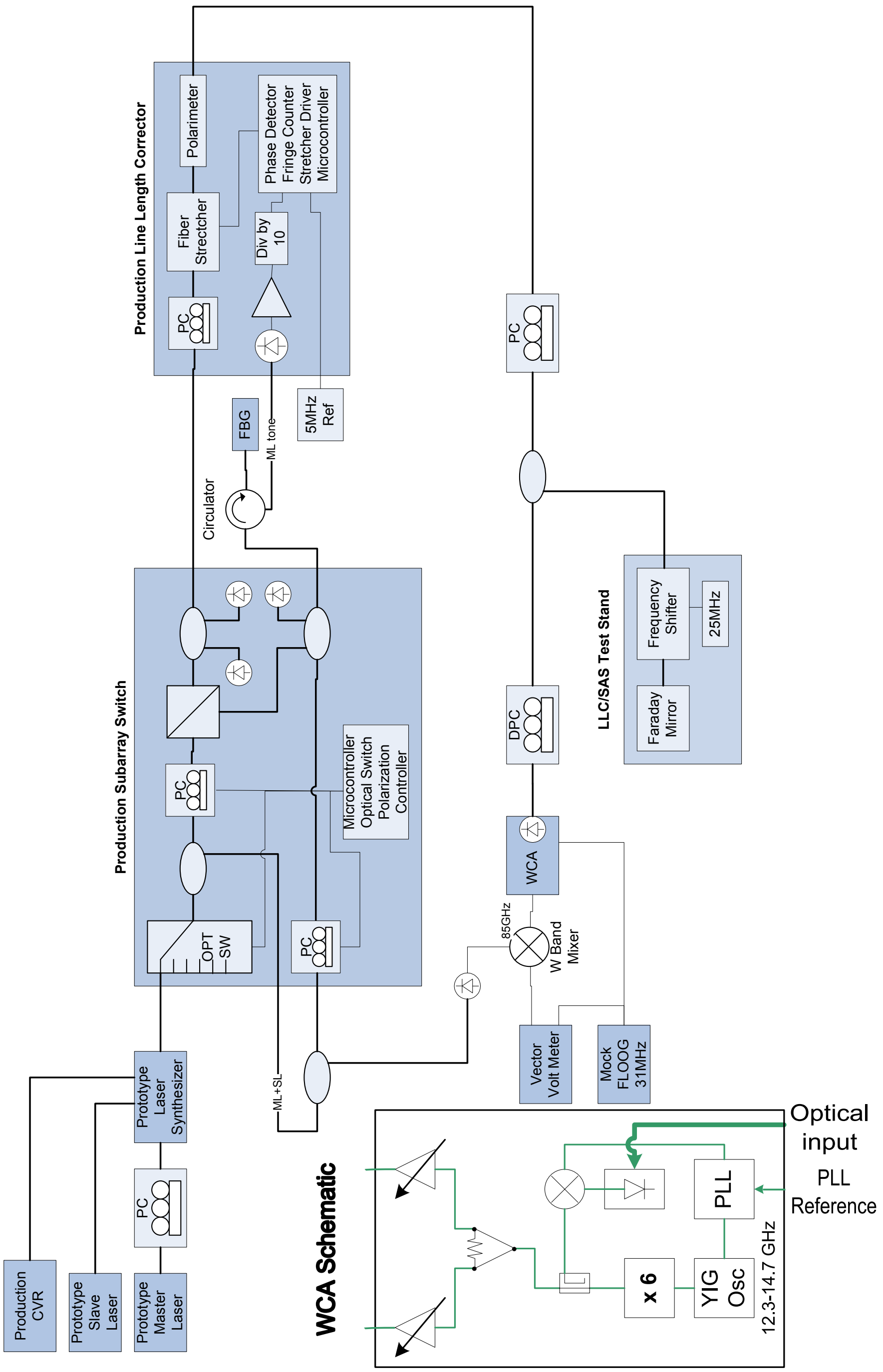


Fig. 24: In this layout test setup is shown, it integrates a basic CLO configuration including a ML, a LS, an EDFA, a SAS; this test set was also used to check the performance of the LLCs and SAS, then a LLC/SAS test stand is in the loop.

The test set used two photomixers, one use as reference and mounted directly in the WCA, and the second was the DUT. It was attached to a w-band mixer and the LO port of this mixer was connected to one of the output ports of the WCA. A 31MHz signal is feed to the PPL reference on the WCA, in consequence the signal produced by the WCA will be 31MHz lower than the one coming from the DUT. Then the mixer intermediate frequency (IF) will be at 31MHz. This signal is then feed to a vector voltmeter (VVM). This instrument measures the phase between the IF and the PLL reference. Then the VVM is capable of measuring any phase change produced by the DUT. By adding a deterministic polarization controller (DPC) the effect of a polarization change on the photomixer phase can be measured.

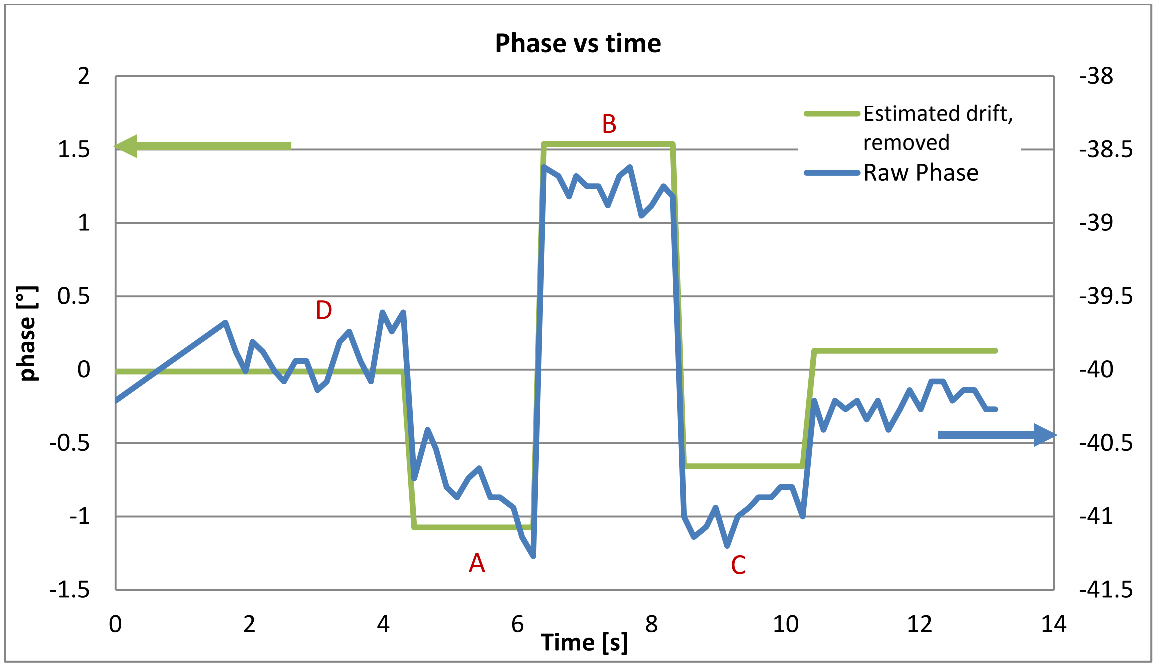


Fig. 25: Plot showing a phase measurement. The red letters indicate the moments in time where the four different polarizations where set, to obtain the data shown in Table 5.

3.3.4 Discussion

The data presented here indicates that the photomixer has a DGD value that in relative terms represents a significant part of the DGD of the complete CLO distribution system. However it was found later that the mechanism that generated this polarization-dependent phase, is produced by a combination of the power-to-phase conversion property of the photomixer, and the PDL of the photomixer. Despite this situation for any practical means this polarization-to-phase conversion in the photomixer behaves exactly like PMD, thus these photomixers must be treated like they have significant and undesired PMD, PDL and power-to-phase conversion.

3.4 Experimental determination of the Fiber stretcher optical model Parameter.

3.4.1 Objective

At the time when this thesis was proposed the Photonics Lab in the Electrical Engineering Department of the Universidad de Chile (DIE) was only capable of making theoretical work. Therefore, having a computer model of the fiber stretcher, which is the most important device of the LLC system of ALMA, was really desirable. This would have had the potential of enabling research progress even when performing laboratory work with the LLC in the DIE was not possible. At the end, this became true. Indeed, it was the existence of this model that allowed the development of the algorithm exposed in chapter 4, which was published later in the IEEE Photonics Journal [32].

3.4.2 Setup

This setup was made taking advantage of the presence of an optical component analyzer from Agilent[11] [12] at the Photonics laboratory of the NRAO technology center for a limited time. This instrument was there for a commercial demonstration.

To obtain the optical model, as shown in Fig. 26, an LLC was mounted on the LLC test stand, which is a custom made device that allows the operation of the LLC and SAS, and at the same time to have troubleshooting accesses to their internal components. Then, the fiber stretcher was connected to the instrument. Finally the fiber stretcher voltage was stepped from 0 to 5 volts and in each step a full range PMD measurement was made. In this process the instrument measures the transmission matrix of the DUT for each wavelength set in the tunable laser and used by the instrument.

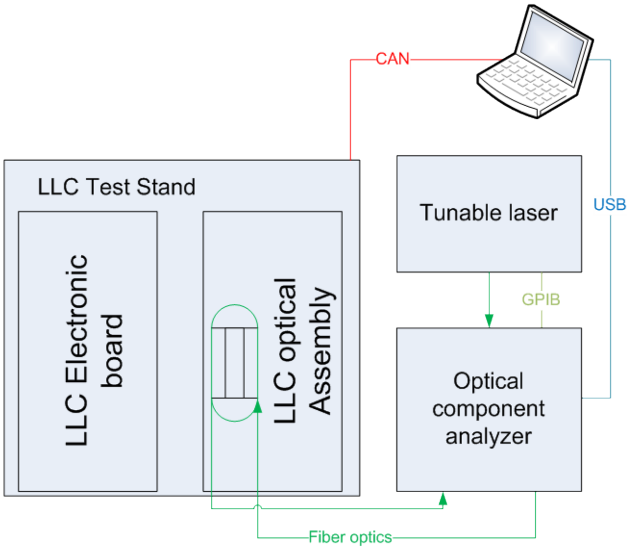


Fig. 26: Test setup used to produce the optical model of the LLC fiber stretcher. The LLC is mounted on the test stand and then the stretcher was connected to the optical component analyzer. The required tunable laser, the test stand and the instrument are controlled by a laptop computer through their respective control interface.

3.4.3 Results and Discussion

As a result a set of transmission matrices was obtained, one for each wavelength and stretcher voltage combination. All of this produces an accurate and realistic optical model of the fiber stretcher. These transmission matrices allow the accurate calculation of the LLC optical output for any given input signal. This can improve the cost and speed of R&D over this optical component of the LLC. Plots like Fig. 29, and the development of the algorithm explained in the chapter 4 are the secondary outcome of this measurement.

[11] This instrument was described in the section 3.1.3.1.
[12] http://www.home.agilent.com/en/pd-1365430-pn-N7788B/optical-component-analyzer?&cc=CL&lc=eng

3.5 Conclusion

In this chapter a summary of several PMD measurement campaigns was made, these campaigns had different impacts on the final outcome of this thesis. The campaign at the VLA was the first approach to a real PMD measurement using standard industrial instruments. Later, the ALMA campaign allowed to preview of the PMD performance of installed fibers on the AOS. This performance was so good that it allowed ALMA to discard the PMD as a risk factor. In consequence, any further PMD measurement was canceled, and this thesis was concentrated on fiber stretcher optimization. The photomixer PMD measurement revealed significant phase-polarization dependence, which can be interpreted as a high PMD of the photomixers.This result was unexpected and added some technical risk to the array phase stability. Finally the determination of the optical model of the fibers stretchers ended up being a really valuable piece of information, allowing the development of the algorithm that is presented in chapter 4.

4 Fiber stretcher polarization optimization

Finding the input state of polarization (SOP) that causes a minimal State of Polarization Change (SOPC) for a fiber optics stretcher is a complex task. The position of these SOP points dubbed "sweet spots", changes with time and can be located at any point on the Poincaré sphere. Maintaining a stable SOP over the entire optical link is important in many applications. One of them is the line length correction (LLC) scheme in the ALMA Photonics Central Local Oscillator (PCLO) system [11][1], which uses a fiber stretcher driven by a phase-lock loop (PLL) to stabilize the optical path length of a photonic signal that is sent to many antennas distributed over distances of up to 15 km (see the Fig. 27). At the antennas, a photomixer is used to receive the mm-wave signal in order to lock the PLL of the local oscillator multiplier chain in the antenna front-end. Any SOPC generated by the fiber stretcher, is converted, through the polarization mode dispersion (PMD) of the buried long fiber and other optical components including the photomixer, into a phase change at the output of the photomixer. To avoid this undesired effect, a polarization controller is used to set the input polarization of the stretcher to one of the minimal-SOPC spots [14][19]. In the current ALMA system, the calibration of the input SOP is performed using an algorithm that iteratively searches for the right input polarization to the stretcher. This algorithm is effective but its convergence time is typically a few minutes and depends on the initial set-point of the fiber stretcher [16]. Since ALMA is a complex system with expensive observing time, every opportunity to speed-up and simplify operations is welcome.

We propose that the position of these sweet-spots can be calculated in a fixed number of SOP measurements using a method, developed by Heffner [36][42] and generalized by Shieh [19], for measuring PMD using Jones Matrix Eigen-analysis (JME).

While Heffner's original method was intended to use measurements of the transition matrix at two different wavelengths, we use an identical algorithm with transition matrices from two different stretching positions of the fiber stretcher. This is justified because Heffner's original method was derived not only for the wavelength regime but more generally for the phase regime, and a phase change can be realized either by a wavelength change or an optical path length change. Besides being faster than the current method, ours is also implementable using the existing LLC hardware, which already includes a non-deterministic polarization controller and, after the stretcher, a polarimeter. This is achieved through a small relaxation of the original requirements of the Jones method for the experimental determination of the transmission matrix, supposing that the uncertainty resulting from the absence of a polarimeter before the stretcher is represented by a constant transformation matrix.

In the rest of this chapter the theoretical background is reviewed as well as the applicability of the algorithm to the ALMA LLC system. Then, we explain the preliminary implementation into the ALMA photonic system, a preliminary measurement and the experimental methodology. This is followed by a description of the technical implementation and the results.

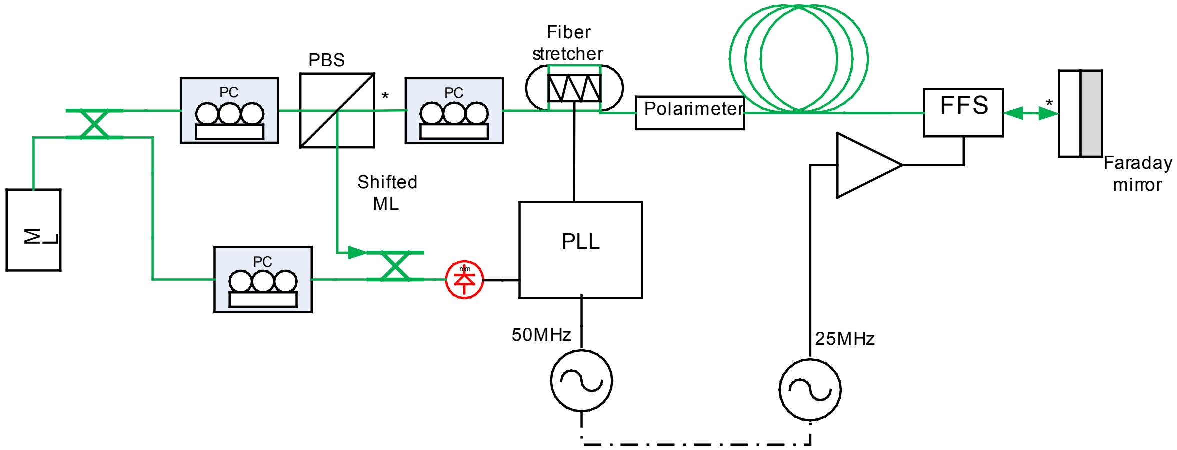


Fig. 27: Simplified diagram of the ALMA line-length correction (LLC) system. A signal generated at the ultra phase-stable master laser (ML) is transported to the Faraday rotator mirror, where it is reflected back, thus crossing the fiber frequency shifter (FFS) twice. As a result the signal acquires a total frequency shift of 50 MHz, and, furthermore, undergoes a Poincaré sphere inversion of its polarization state. The signal returning to the polarizing beam splitter is then mixed with a direct ML signal to produce a beat note at 50 MHz. This signal is used to lock the PLL that drives the fiber stretcher, effectively stabilizing the total length between asterisks. If a SOPC of the optical signal is produced by the stretcher, due to the PMD of the rest of the fiber line and other optical components such as the photomixer (not shown in this figure), a phase change of the signal at the antenna will be produced, thus possibly rendering the antenna phase out of the phase stability specification. Therefore, just before the input of the fiber stretcher, a polarization controller (PC) is inserted. The PC is used together with the polarimeter to find the fiber stretcher sweet-spots.

4.1 Theoretical Background

4.1.1 Description of the Method

The method presented here, as well as the old algorithm, is based on the empirical observation that the SOP output of the stretcher for a constant SOP input rotates around the axis through both sweet-spots (Fig. 28) when the fiber stretcher is actuated. This is the same effect found in the polarization mode dispersion (PMD), where the SOP rotates around the axis through the principal states of polarization (PSP) when a monochromatic wave is swept in frequency through a bi-refringent fiber [34].

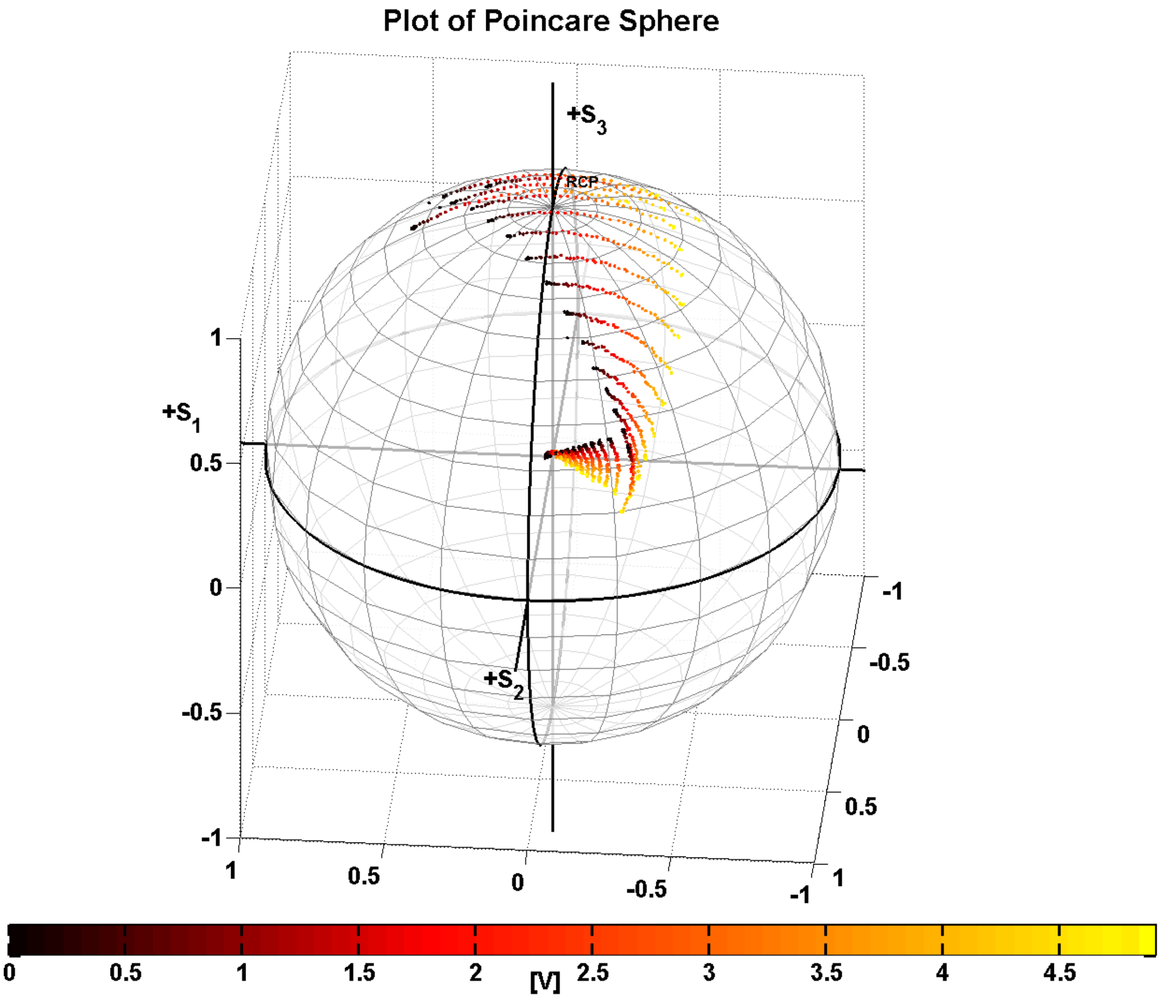


Fig. 28: Poincaré sphere showing the polarization trajectory of a stretcher while being optimized using the iterative baseline algorithm. It can be seen that the polarization tends to rotate around the finally-found "sweet- spot" after many sweeps of the stretcher. In the plot the color bar indicates the piezoelectric actuator voltage of the stretcher in a given moment, during the process the SOP is progressively moved from a position where the sweeps of the stretcher voltage produce big arcs over the sphere to a position where this arcs are minimized. In this view, the sweet-spot is hovering over the origin of the Cartesian axes at the center of the view.

The explanation of this similarly is that a wavelength change is equivalent to a change of the fiber length by mechanical stretching. In both cases the ratio of the optical path length to the wavelength is changing the same way and therefore the effect on the output phase is the same. This is only true if the stretching range is small enough (linear approximation), so that the fiber cross-section and its elliptical distortions in the bends of the stretcher windings are assumed not to change. Under this assumption, we would see the SOP describing a non-circular path over the Poincaré sphere. Furthermore, measured PSPs (Eigenstates with respect to wavelength) and sweet-spots (Eigenstates with respect to length change) should be identical if the fiber length affected by the stretching is the complete fiber between the polarization controller and the polarimeter. However, due to instrumental reasons the stretched section is always shorter than the complete fiber, preventing the experimental verification of the similitude between sweet-spots and PSP. Just for clarity we will keep calling the PSPs "sweet spots" when they are measured using a length change instead of a wavelength change, as in the new algorithm.

The calculation required to measure the PMD vector using this scheme was formulated by Heffner [36][42], and is based on the determination of the rotation of the SOP around the axis through the PSPs. As previously explained we propose that this algorithm can be used analogously to calculate the position of the sweet-spots by replacing the parameter of the frequency change by the stretcher state change, a principle which was proposed already earlier in a more generalized way [19]. The final method then consists of measuring the transmission matrix T at two different stretcher positions yielding $T(L)$ and $T' = T(L + \Delta L)$, where L is the fiber length, and the sweet-spots are given by the Eigenvectors of the product matrix $T' \, T^{-1}$.

For an experimental determination of the two forward transmission matrices T of an unknown, but linear, time-invariant optical device, according to Heffner the algorithm of Jones is used [35]. A laser followed by a polarization controller generates a linearly polarized wave parallel to the x-axis and the output polarization is measured by a polarimeter, resulting in the Jones vector h. Then, signals polarized parallel to the y-axis and the bisector axis between x and y are transmitted, giving the vectors v and q respectively. The transmission matrix can then be calculated as in (5) and (6):

4.1.2 Effect of PDL

Just as in Heffner's scheme, a non-orthogonality of the calculated sweet-spots can be produced by a polarization dependent loss (PDL). In the case of a fiber stretcher, the PDL should be kept very small by design [43] and, in fact, this is valid for the whole ALMA photonic system. Therefore, such an effect can be neglected here.

4.1.3 Effect of components between the PC and the stretcher

The proposed method requires the production of specific and precise input polarizations to the fiber stretcher. Unfortunately the current hardware of the ALMA LLC system only has a piezoelectric polarization rotator based on fiber squeezers at the input and a polarimeter at the output of the stretcher, making it impossible to control precisely the polarization state delivered to the stretcher. However, there is a way around this problem.

To model this uncertainty, let us suppose that we have a precise way to set the required input polarization, but between this and the stretcher there is an unknown polarization-changing device with a transfer matrix T_u that does not change with time or the stretcher position. Then, if the method is applied two transmission matrices will be obtained, T_k', and T_k, each of them containing the matrix T_u as.

$$T_u' = T_u'; T_k = TT_u \qquad \textbf{(14)}$$

Then:

$$T'_k T_k^{-1} = \left(T' T_u\right)(T T_u)^{-1} = T' T_u T_u^{-1} T^{-1} = T' T^{-1} \quad \textbf{(15)}$$

As it is shown in (15), the uncertainty on the input polarization is not important for the method, unless it changes during the measurement. This means that the method can be applied by only measuring the output polarization and keeping the relationship between the output polarization vectors consistent with the Jones method, which means two of them need to be mutually orthogonal and a third located half way between the formers. With this, from the point of view of the measurement and actuation, the current hardware is sufficient to apply the method to the ALMA LLC System.

4.1.4 First experimental validation through measurement and simulation

As a preliminary validation of the concept before its full implementation into a large group of telescopes, the algorithm was run with a single fiber stretcher and the results were tested using a simulation in Matlab. This simulation was performed using a data set of transmission matrices obtained from a production fiber stretcher of the ALMA photonic system, using an Agilent optical component analyzer, an instrument that uses JME to measure PMD in optical components[13]. One part of its output is the Jones transmission matrices at different wavelengths used to calculate the PMD of the component. The instrument has a deterministic polarization controller to set the required input polarizations for the device under test (DUT) and a polarimeter to measure the output polarizations of the DUT. The light source was an Agilent tunable external cavity diode laser.

[13] For more details about the capture method of this data set please check the page 48

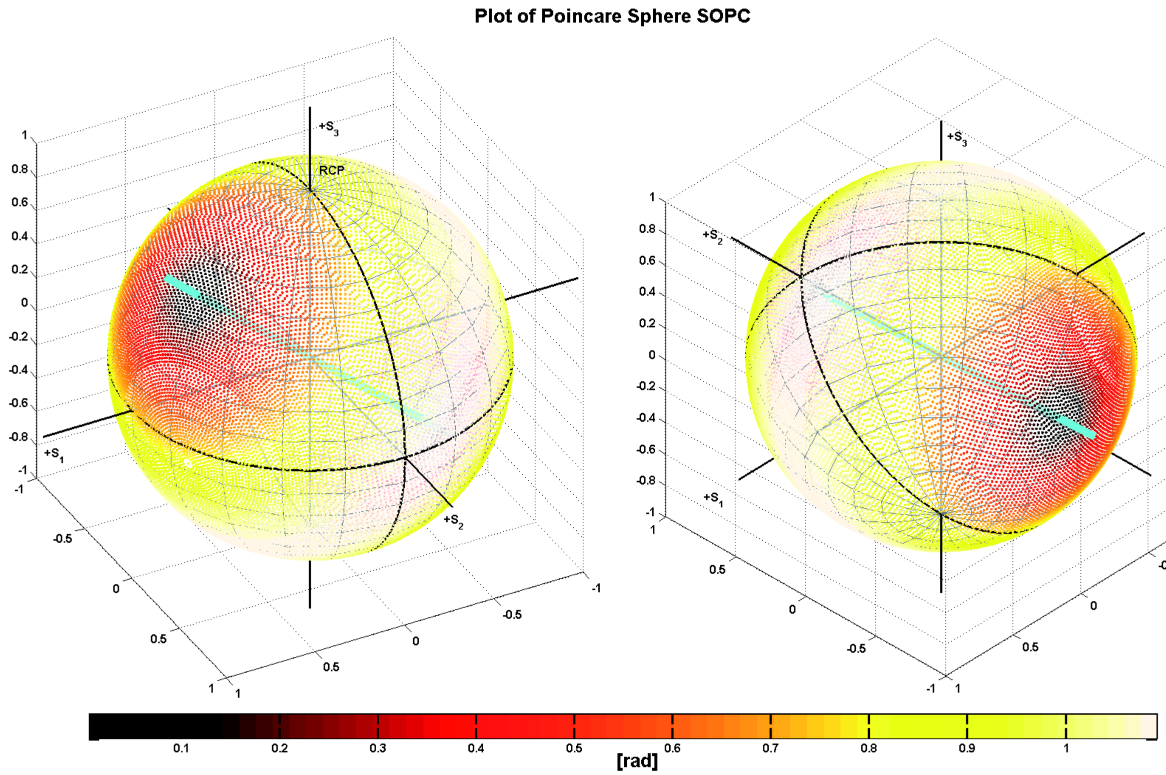


Fig. 29: Poincaré spheres of the calculated SOPC produced by a fiber stretcher actuation, showing two orthogonal sweets-spots of a fiber stretcher (the darker areas, scale in radians). It also shows the calculation of sweet-spots with the proposed method, as thick vectors. There is a good agreement between the exhaustive method and the positions calculated by the proposed here.

To generate the plot shown in Fig. 29, the transmission matrices obtained with this instrument at 1556 nm were processed with a Matlab script that calculated the SOPC between two respective stretcher positions with input polarizations covering the whole Poincaré sphere. A color code is used to indicate the magnitude of the SOPC as a function of the input polarization. This plot clearly shows two orthogonal low-SOPC areas. Superimposed are the two sweet-spot vectors calculated by the proposed algorithm. The SOPC is defined by (1)

4.2 Implementation of the new algorithm

4.2.1 Technical Aspects

The algorithm was implemented as a python script running in the ALMA control computers. Since it uses only 41 multiplications and 20 additions, it is possible to implement it at the firmware level on any microcontroller. The final implementation will depend on the current usage of processor resources in the LLC microcontroller.

A point that requires special attention is how to precisely control the SOP at the output of the stretcher, using the PC at the input. This PC has as control input 4 voltages used to drive 4 squeezers that effectively rotate the output SOP around an axis. The control strategy for this PC is simple. One voltage at a time is changed using a fixed step size to find the minimum angular distance between the current and the desired SOP. When the distance stops decreasing and starts increasing, the step change direction and it size is divided by two. If the distance change between two voltage steps is smaller than an absolute fixed threshold, or if it is smaller than a certain fraction of the current distance to the desired SOP , the method will move on to the next squeezer. It was found that 1/40 of the current SOPC works well for the second threshold. In this way the method will give only a coarse optimization to squeezers that at best can change the SOP only a little towards the desired one, and give a more precise and time consuming optimization to those that can change the SOP more closely to the required SOP, for any starting SOP. The algorithm cycles through the 4 squeezers in quick succession until the distance from the desired SOP is smaller than a second threshold which is chosen as a compromise between short execution time and the precision that polarization control is capable of achieving, which in ALMA was set to 0.009 radians. It was found that this method always converges regardless of the starting SOP.

In order to reduce the execution time of the algorithm, the SOPs to be set for the measurement are selected as follows.

Whatever the SOP is at the beginning of the algorithm; it will be the first of the three SOP h given in Stokes space using the Cartesian coordinates of the Poincaré sphere. The second SOP will be $v = -h$ and the last one, q, will be selected by replacing the smallest Cartesian coordinate of h by minus the sum of the squares of the other two coordinates, divided by the coordinate to be replaced. For example, if S_1 is the smallest coordinate then it shall be replaced by S_1':

$$S_1' = \frac{-(S_3^2 + S_2^2)}{S_1} \qquad \textbf{(16)}$$

This is repeated for the other cases. If the smallest coordinate is 0, it is replaced by 1 and the other two are set to 0. In addition we must make sure that none of the vectors, h, v, q is parallel to the axis x, because if that were the case, the calculation of the transmission matrix T would produce a division by 0 as can be seen in (6). It is also recommended that none of them be close to parallel to the x axis in order to reduce the impact of the measurement noise, how far away will depend on the noise of the SOP measurements. If the previous condition is not accomplished, then the vectors to use will be defined by (17)

$$h = [1,0,0,1]; q = [1,0,1,0]; v = [1,0,0,-1] \qquad \textbf{(17)}$$

Another solution is to change the reference frame to avoid the division by 0, but this would increase the complexity of the method.

Although the proposed algorithm has no dependence on the starting point, this particular implementation has some degree of dependence, because of the design decision of using the current SOP as a seed to choose the SOPs required by the algorithm for the calculations. This decision reduces, on average, the time required to perform the calibration by 25%.

4.2.2 Experimental Methodology

For the algorithm performance test, two python scripts were written to perform recurrent runs of the baseline algorithm and the proposed one, using the maximum number of LLCs that was possible each time. Before each run the starting SOP was randomized.

The LLCs used for the test were located at the array operations site of ALMA located on the Chajnantor plateau at 5000 m above sea level. The tests were conducted during engineering time, for this reason the number of runs was limited to 50. For each data set, the test was run using every available LLC. However, it was not always possible to have the same set of LLCs on different days or for sufficient times in sequence to ensure a valid comparison. The one presented here was made using only the LLCs that were present and had completed 50 runs in both data sets.

For these comparisons of the performance of both algorithms, the SOPC is measured using the internal measurement routine of the LLC. This routine sweeps the stretcher position, while the SOPC is monitored. The reported values are the maximum SOPCs detected during this process.

4.2.3 Experimental Results

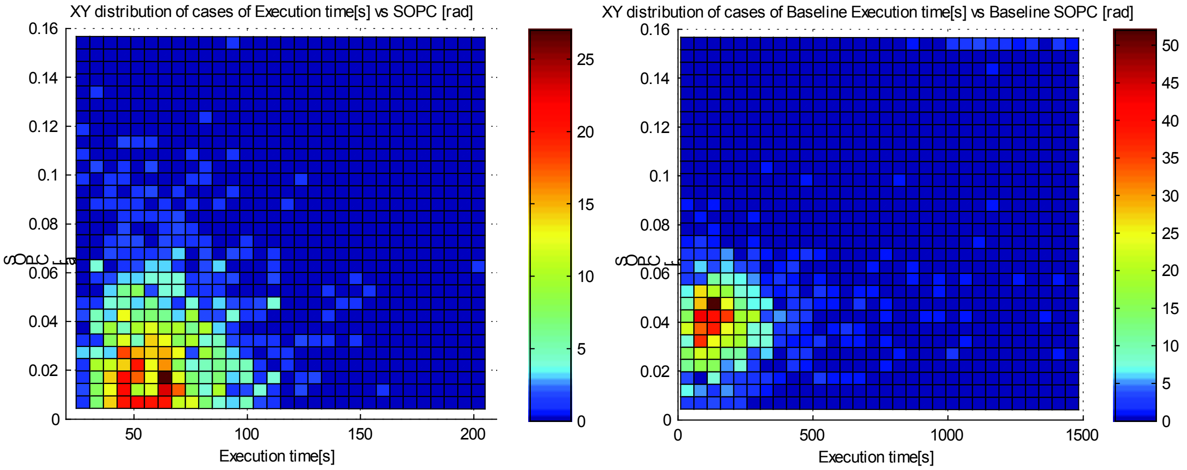


Fig. 30: Color XY distributions of case counts, which shows the interdependence between execution time and final SOPC. The SOPC axis is limited to 0.16 radians to allow easier comparison between the algorithms. A consequence of this is that all data points $> 0.16rad$, are accumulated in this grid row, mostly seen in the plot for the baseline algorithm (right panel) the left panel shows the results of the new algorithm.

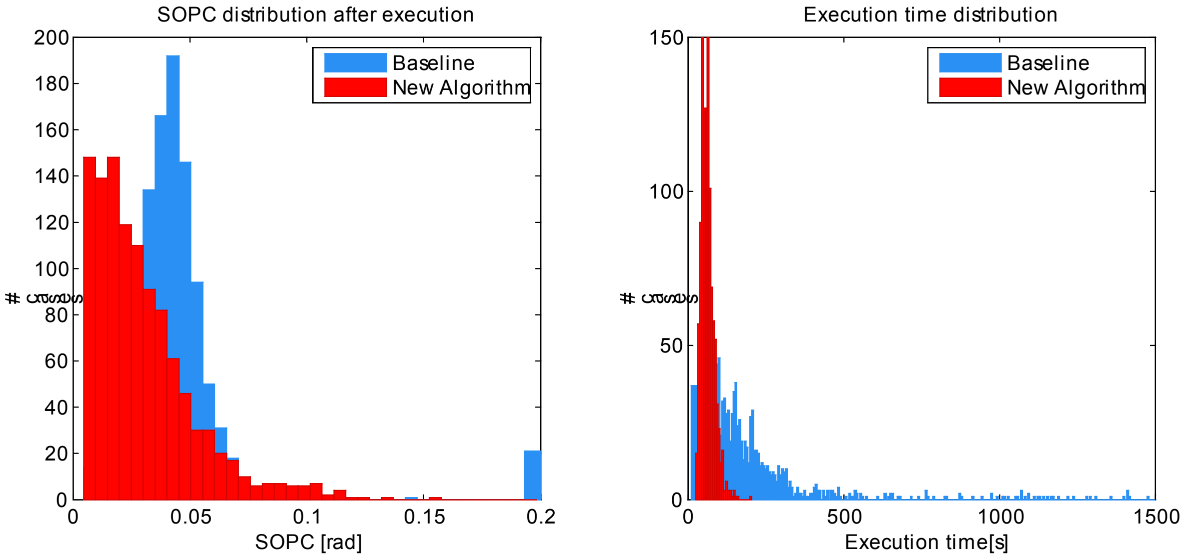


Fig. 31: Histograms showing the execution time and the SOPC after each execution. The left panel shows the SOPC after the execution of both algorithms, and the right plot shows their execution time distribution. As mentioned in section 2, the SOPC is measured by sweeping the fiber stretcher across its complete range and measuring the polarization change during this process; the reported value is the maximum measured change. The highest bin of the SOPC plot accumulates the values bigger than 0.2 radians.

Table 6: Compared performance statistics between baseline and the proposed algorithm

Algorithm	$\bar{x}$ Exec time [s]	σ Exec. time [s]	$\bar{x}$ Remaining SOPC [rad]	σ Remaining SOPC [rad]
Proposed	62.5	21.1	0.0297	0.0218
Baseline	211.5	219.8	0.0583	0.1409

Following the methodology explained in the section 4.2.2, Figures Fig. 30 and Fig. 31 were produced. They show the performance of both the base line and the proposed algorithm, as a function of the execution time and the final SOPC achieved after execution of each algorithm. In both cases a smaller number indicates better performance. The performance indicators for each of the methods are presented in

Table 6. This results show that the new algorithm is superior to the baseline in every aspect. Fig. 30 shows that both algorithms have no correlation between execution time and residual SOPC, Fig. 31 shows a graphical comparison of both performance parameters of the used as benchmark. This clearly shows how the new algorithm with high probability will produce a more accurate calibration and in far shorter time.

4.3 Conclusion

In this chapter a method to calculate the minimal SOPC input polarization on fiber stretchers has been presented. We have demonstrated that characterizing the fiber stretcher at two positions is sufficient to find the position of the sweet-spots. A good agreement of the sweet-spots found in this way with a characterization of the SOPC through the whole Poincaré sphere has been shown using a simulation. The developed algorithm based on this has proven to be simple, fast, and accurate.

A field test was conducted using 50 LLCs, showing that with the help of a simple polarization control algorithm, the proposed method reduces the average time to finish the LLC calibration process by 61% and the standard deviation of this time by 90%. It also reduces the remaining average SOPC after optimization by 50% and its standard deviation by 85%.

The equivalence between a wavelength step and a fiber length change produced by a well-designed stretcher has been explained, and we show that the sweet-spots and the PSPs are conceptually the same. The reported algorithm may be used in fiber stretcher applications where good polarization stability is required and time constraints exist, for example, an environment that requires permanent recalibration of the stretcher polarization input, thereby producing a significant reduction in the accumulated calibration time.

5 Discussion and Further Work

The aim of this thesis was to investigate the details of ALMA photonic central local oscillator, with the hope of finding a space for contribution. This is especially challenging if you consider that ALMA is an observatory that has been on the design bench for more than 30 years, as the first ALMA memo[14] was issued in 1982.

The first suggestion for research was the PMD performance of the fiber optic network that had to be installed on the AOS for the distribution of the CLO to the antennas. The risk that the final PMD of the fibers would be too high to allow the CLO transmission system to achieve the specified performance was causing many sleepless nights to the photonics team. Then, a complete study of the system design, and its interaction with PMD was made, putting special attention to how PMD affects the performance of the system. This was then followed by a study of the relevant PMD measurement and compensation techniques. Then a second area of research showed up: The component of the CLO that has most interaction with the PMD and which is the very center of the CLO distribution system is the LLC. Thus, finding ways to improve the operation of this device became a second line of research. After the first PMD measurement campaign at the AOS it became clear that the risk of having too high PMD of the installed fibers was significantly less than assumed and the center line of research became the LLC. But all the previous work was not lost; it was the understanding of the PMD and its different measurement techniques that allowed the confirmation of the previous suspicion of the photonics group, that the fiber stretcher sweet spots and PSPs are the same conceptual thing. The confirmation of this enabled finally the development of the new LLC polarization optimization algorithm based on JME. But the implementation of this algorithm on the ALMA hardware presented a new challenge, as the algorithm required the precise control of the input SOP to the fiber stretcher, with the only polarimeter in the system being at the output of the fiber stretcher. To solve this, the method was adapted to relax the JME input polarization requirement converting it into only a relative position between the SOPs at the output of the fibers stretcher. Also, a method was developed that allows the precise control of the output SOP of the fiber stretcher, making the input polarization controller and the output polarimenter working together to control the precise output SOP. This was done without making any assumption about the fiber stretcher transmission matrix.

The testing of the algorithm proposed with the ALMA production hardware showed a 70% reduction of the average calibration time, and an improvement of the final accuracy, reducing the average residual calibration error by half. This new algorithm can be applied to eliminate or reduce undesired polarization change effects in any device where the polarization is rotated around an axis when a parameter like temperature, voltage, pressure, magnetic field, etc. is changed. This concept was dubbed dynamic eigenstates of polarization by Shieh in 2001 [19].

Despite the effort put into the development of this thesis, long-term PMD measurements of the longer fiber links at the AOS are still pending. They could be performed after fiber

[14] The ALMA memo series, is a big set of working documents about different aspects of ALMA, since indicial concept to very detailed descriptions of ALMA components and specifications. They can be found in https://science.nrao.edu/facilities/alma/aboutALMA/Technology/ALMA_Memo_Series.

installation at the AOS is completed. There are opportunities to apply parts of the proposed algorithm in other devices within the CLO, like the power optimization of the return signal at the SAS. On the other hand, and since in the current algorithm the most time consuming step is the setup of the precise SOPs required for the calculation, the fast algorithm presented here could be made even faster by developing a new method that can calculate the PSP using a random set of SOPs with no relative position requirement beyond being spread more or less evenly over the Poincare sphere. Finally potential applications of this algorithm on systems not related to ALMA should be explored. These potential applications should be investigated in any system where a dynamic eigenstate of polarization can be found, one example of this can be the development of low cost fiber optics temperature transducers.

6 Bibliography

[1] J.-F. Cliche and B. Shillue, “Precision timing control for radioastronomy: maintaining femtosecond synchronization in the Atacama Large Millimeter Array,” *Control Systems Magazine, IEEE*, vol. 26, no. 1, pp. 19–26, 2006.

[2] M. Engineering, “SITE-20.04.01.09-021-D.DWG,” *private communication*, 2013.

[3] M. Engineering, “SITE-20.04.01.09-022-D.DWG,” *private communication*, 2013.

[4] M. Engineering, “SITE-20.04.01.09-023-D.DWG,” *private communication*, 2013.

[5] S. M. Foreman, K. W. Holman, D. D. Hudson, D. J. Jones, and J. Ye, “Remote transfer of ultrastable frequency references via fiber networks,” *Review of Scientific Instruments*, vol. 78, no. 2, p. 021101, Feb. 2007.

[6] I. Coddington, W. C. Swann, L. Lorini, J. C. Bergquist, Y. Le Coq, C. W. Oates, Q. Quraishi, K. S. Feder, J. W. Nicholson, P. S. Westbrook, S. A. Diddams, and N. R. Newbury, “Coherent optical link over hundreds of metres and hundreds of terahertz with subfemtosecond timing jitter,” *Nature Photonics*, vol. 1, no. 5, pp. 283–287, 2007.

[7] J.-F. Cliche, B. Shillue, M. Tetu, and M. Poulin, “A 100-GHz-tunable photonic millimeter wave synthesizer for the Atacama Large Millimeter Array radiotelescope,” in *Microwave Symposium, 2007. IEEE/MTT-S International*, 2007, pp. 349–352.

[8] L. D’Addario, E. Bryerton, R. Sramek, W. Shillue, S. Thacker, and D.-. M. R. -feb, “LOCAL OSCILLATORS,” in *ALMA Project Book*, 2001, pp. 7.1–7.36.

[9] K. Sato, T. Hara, S. Kuji, K. Asari, M. Nishio, and N. Kawano, “Development of an ultrastable fiber optic frequency distribution system using an optical delay control module [for frequency standard and VLBI],” *IEEE Transactions on Instrumentation and Measurement*, vol. 49, no. 1, pp. 19–24, 2000.

[10] F. Narbonneau, M. Lours, S. Bize, A. Clairon, G. Santarelli, O. Lopez, C. Daussy, A. Amy-Klein, and C. Chardonnet, “High resolution frequency standard dissemination via optical fiber metropolitan network,” *Review of scientific instruments*, vol. 77, p. 064701, 2006.

[11] L. S. Ma, P. Jungner, J. Ye, and J. L. Hall, “Delivering The Same Optical Frequency at 2 Places - Accurate Cancellation of Phase Noise Introduced By An Optical-Fiber or Other Time-Varying Path,” *Optics Letters*, vol. 19, no. 21, pp. 1777–1779, 1994.

[12] N. R. Newbury, P. A. Williams, and W. C. Swann, “Coherent transfer of an optical carrier over 251 km,” *Optics Letters*, vol. 32, no. 21, pp. 3056–3058, 2007.

[13] N. Radio and A. Observatory, "ALMA Memo # 443 ALMA LO Distribution Round Trip Phase Correction Author : Bill Shillue National Radio Astronomy Observatory 2 ALMA LO Distribution System Description," *System*, 2002.

[14] P. Shen, A-4650-2012, W. P. Shillue, and S. Albanna, "The Temporal Drift Due to Polarization Noise in a Photonic Phase Reference Distribution System," *Journal of Lightwave Technology*, vol. 26, no. 15, pp. 2754–2763, 2008.

[15] B. Shillue, S. AlBanna, and L. D'Addario, "Transmission of low phase noise, low phase drift millimeter-wavelength references by a stabilized fiber distribution system," in *IEEE International Topical Meeting on Microwave Photonics (IEEE Cat. No.04EX859)*, 2004, pp. 201–204.

[16] J. Cliche and B. Shillue, "Central LO Article Second Delivery (CLOA2) PAS Test Results," *private communication*, 2011.

[17] S. Chip, "An analysis on the Effect of Dual Wavelength Polarization Alignment on the Accuracy of Recovered Beat Note Phase." 2004.

[18] P. Shen, N. J. Gomes, P. A. Davies, W. P. Shillue, and P. G. Huggard, "PMD Effects on the Analogue Signal Transmission." 2004.

[19] W. Shieh and H. Kogelnik, "Dynamic eigenstates of polarization," *IEEE Photonics Technology Letters*, vol. 13, no. 1, pp. 40–42, 2001.

[20] T. Kudou, M. Iguchi, M. Masuda, and T. Ozeki, "Theoretical basis of polarization mode dispersion equalization up to the second order," *Journal of Lightwave Technology*, vol. 18, no. 4, pp. 614–617, Apr. 2000.

[21] F. Heismann, D. A. Fishman, and D. L. Wilson, "Automatic compensation of first order polarization mode dispersion in a 10 Gb/s transmission system," 1998, vol. 1, pp. 529–530 vol.1.

[22] H. Sunnerud, M. Karlsson, Chongjin Xie, and P. A. Andrekson, "Polarization-mode dispersion in high-speed fiber-optic transmission systems," *Lightwave Technology, Journal of*, vol. 20, no. 12, pp. 2204–2219, 2002.

[23] C. Francia, F. Bruyere, J. P. Thiery, and D. Penninckx, "Simple dynamic polarisation mode dispersion compensator," *Electronics Letters*, vol. 35, no. 5, pp. 414–415, 1999.

[24] H. Rosenfeldt, C. Knothe, R. Ulrich, E. Brinkmeyer, U. Feiste, C. Schubert, J. Berger, R. Ludwig, H. G. Weber, and A. A.-E. Ehrhardt, "Automatic PMD compensation at 40 Gbit/s and 80 Gbit/s using a 3-dimensional DOP evaluation for feedback," 2001, vol. 4, pp. PD27–1–PD27–3 vol.4.

[25] O. Karlsson, J. Brentel, and P. A. Andrekson, “Long-term measurement of PMD and polarization drift in installed fibers,” *Lightwave Technology, Journal of*, vol. 18, no. 7, pp. 941–951, 2000.

[26] A. B. Shillue and S. Albanna, “ALMA Memo # 507 ALMA First LO Reference : Elimination of Large Phase Fluctuations due to lightwave polarization effects,” pp. 1–34, 2004.

[27] H. Kiuchi and M. Ishiguro, “An alternative scheme of round-trip phase correction.” Tokyo, Japan, 2002.

[28] P. A. Williams, “Modulation phase-shift measurement of PMD using only four launched polarisation states: a new algorithm,” *Electronics Letters*, vol. 35, no. 18, pp. 1578–1579, 1999.

[29] B. Shillue and D. Thacker, “Evaluation of the NAOJ Alternative LO System,” 2007.

[30] H. Kiuchi, “Alternative Scheme Line Length Corrector Prototype Test Report Subsystem Laboratory Test,” 2007.

[31] P. Shen, N. J. Gomes, P. A. Davies, and W. P. Shillue, “Polarization Mode Noise in Ultra-low Drift Phase Reference Distribution System over a Fiber Network,” in *2005 International Topical Meeting on Microwave Photonics*, 2005, pp. 297–300.

[32] J. A. Castillo and E. A. Michael, “Fast Algorithm to Find the Minimal Polarization Change Points on a Fiber-Optic Stretcher Using One Polarization Rotator and a Polarimeter,” *IEEE Photonics Journal*, vol. 4, no. 6, pp. 2390–2397, Dec. 2012.

[33] J. A. Castillo and E. A. Michael, “Practical implementation and on the field test results of an enhanced algorithm for polarization optimization of the line length corrector of ALMA central local oscillator,” in *SPIE 8452*, 2012, p. 84522N–84522N.

[34] P. Williams, “PMD measurement techniques and how to avoid the pitfalls,” *Journal of Optical and Fiber Communications Reports*, vol. 1, no. 1, pp. 84–105, Sep. 2004.

[35] R. C. Jones, “A new calculus for the treatment of optical systems VI. Experimental determination of the matrix,” *Journal of the Optical Society of America*, vol. 37, no. 2, p. 110, Feb. 1947.

[36] B. L. Heffner, “Automated measurement of polarization mode dispersion using Jones matrix eigenanalysis,” *Photonics Technology Letters, IEEE*, vol. 4, no. 9, pp. 1066–1069, 1992.

[37] C. R. Menyuk and A. Galtarossa, *Polarization mode dispersion*, vol. 1. Springer, 2005.

[38] N. Cyr, “Polarization-mode dispersion measurement: Generalization of the interferometric method to any coupling regime,” *Journal of lightwave technology*, vol. 22, no. 3, pp. 794–805, 2004.

[39] B. L. Heffner, “Influence of optical source characteristics on the measurement of polarization-mode dispersion of highly mode-coupled fibers,” *Optics Letters*, vol. 21, no. 2, p. 113, Jan. 1996.

[40] P. Williams and A. Barlow, “Narrowband measurements of polarization-mode dispersion using the modulation phase shift technique,” *NIST special …*, vol. 930, pp. 23–26, Sep. 1998.

[41] C. Jacques and H. Kurlandczy, “AOS Fibre Test Plan,” *private communication*, pp. 1–16, 2009.

[42] B. L. Heffner, “Accurate, automated measurement of differential group delay dispersion and principal state variation using Jones matrix eigenanalysis,” *Photonics Technology Letters, IEEE*, vol. 5, no. 7, pp. 814–817, 1993.

[43] W. Grammer, G. Stahlman, E. Ford, and H. Kurlandczyk, “Line Length Corrector LRU Fiber Stretcher Assembly Technical Specifications Status : Released,” 2010.

Appendices

Python Scripts

The following scripts were used to implement and test the algorithm exposed in the 4th section.

JLLC.py:

Line length corrector class extension that implements the algorithm

```
import sys, numpy, copy
from pylab import matrix,cross,arccos,pi,array
from struct import pack, unpack
from time import sleep,time
from CCL.LLC import LLC
from StokesJones import *
from transmission import *
import lib.BEDefinitions as Bdef


class JLLC(LLC):
    def __init__(self, \
                antennaName = None, \
                componentName = None, \
                stickyFlag = False):

 #       try:
        LLC.__init__(self, \
               antennaName, \
               componentName, \
               stickyFlag)
  #      except Exception, e:
   #         print 'LLC component is not running \n' + str(e)
        self.Calibration_matrix=[]
        self.averageN=0
        self.__antennaName__ = antennaName
        self.hwError = False
    def hwOperational(self):
        LLC.hwOperational(self)
        self.Calibration_matrix=self.get_cal_matrix()
    def get_cal_matrix(self):
        '''This function gets the Calibration matrix for the LLC polarimetr
        it uses as input the LLC object'''
        M=[]
        for i in range(0,4):
            fila=[]
            for j in range(0,4):
                numero=[]
                for k in range(0,4):
                    #Calcula direccion a leer partindo por 0x100
                    adress=0x100+i*16+j*4+k
                    self.SET_READ_FRAM([adress/(2**8),adress & 0x00ff])
                    numero.append(self.GET_FRAM_BYTE()[0])
                d=pack('4B',numero[3],numero[2],numero[1],numero[0])
                elemento=unpack('f',d)[0]
                fila.append(elemento)
            M.append(fila)
        return matrix(M)
    def get_SOP_normalized(self, averaging=False):
        '''This function return the Stokes parameters measured by the LLC 
polarimeter
        it uses as input the LLC object and the calibration matrix'''
        if self.Calibration_matrix==[]:
            self.Calibration_matrix=self.get_cal_matrix()
        if averaging:
            self.averageN=self.averageN+1
            Voltages_polarim=[]
            for i in range (10):
                
Voltages_polarim.append(matrix([self.GET_POL_MON1()[0],self.GET_POL_MON2()[0],
self.GET_POL_MON3()[0],self.GET_POL_MON4()[0]]))
                sleep(0.05)
            v = array(Voltages_polarim)
            Voltages_polarim = matrix(v.mean(axis=0))
        else :
            
Voltages_polarim=matrix([self.GET_POL_MON1()[0],self.GET_POL_MON2()[0],self.GE
T_POL_MON3()[0],self.GET_POL_MON4()[0]])
        SOP=self.Calibration_matrix*Voltages_polarim.transpose()
        sop=SOP/(SOP[1,0]**2+SOP[2,0]**2+SOP[3,0]**2)**.5
        sop=matrix([SOP[0,0],sop[1,0],sop[2,0],sop[3,0]])
        return sop.transpose()

    def pola_optimization(self,Low_Volt=0.2,High_Volt=4.5,thres=0.009):
        ''' optimizes the llc SOP by using the method created by Jorge 
Castillo
        '''
        self.SET_VF_CTL(1)
        output1=[]
        output2=[]
        output3=[]


 #       print ('valores a utilizar',Low_Volt,High_Volt)
        #the stretcher is set to 0V
        self.SET_VF_O(Low_Volt)

        sleep(0.05)
        output1.append(self.get_SOP_normalized(True).transpose())
        orto = self.SOP_retriver(output1[0].transpose())

        #self.control_pola(vector)

        #retriever the normalizedd SOP for the stretcher at 0V
        self.SET_VF_O(High_Volt)
        sleep(0.05)
        output1.append(self.get_SOP_normalized(True).transpose())
        #retriver the normalizedd SOP for the stretcher at 5V
        #the stretcher is set to 0V for the second polarizing vector
        self.SET_VF_O(Low_Volt)
        sleep(0.05)
        vector=orto[1]
        self.control_pola(vector,thes=thres)
        output2.append(self.get_SOP_normalized(True).transpose())
        self.SET_VF_O(High_Volt)
        sleep(0.05)
        output2.append(self.get_SOP_normalized(True).transpose())
        self.SET_VF_O(Low_Volt)
        sleep(0.05)
        vector=orto[2]
        self.control_pola(vector,thes=thres)
        output3.append(self.get_SOP_normalized(True).transpose())
        self.SET_VF_O(High_Volt)
        sleep(0.05)
        output3.append(self.get_SOP_normalized(True).transpose())
        aux1=list_to_matrix(output1)
        aux2=list_to_matrix(output2)
        aux3=list_to_matrix(output3)
        aux =list_to_matrix1([aux1, aux2, aux3])
        jones = Stokes2Jones(aux)
        transmission0V = Transmission_matrix(jones[0,:],jones[2,:],jones[4,:])
        transmission5V = Transmission_matrix(jones[1,:],jones[3,:],jones[5,:])
        transmission0V = numpy.linalg.inv(transmission0V)
        transmission = transmission5V*transmission0V
 #       print output1, output2, output3
        if ~isnan(transmission).any():
            D,V = linalg.eig(transmission)
        else:
            sys.stdout.writelines(Bdef.RED+' math error happen on llc '+
self.__antennaName__ +Bdef.NO_COLOR+'\n')
            self.__randomize_P__()
            return NaN,NaN
        stokes = Jones2Stokes(V[:,0])
        stokes = stokes/stokes[:,0]


#========================================================================
        # stokesD = Jones2Stokes(V[:,1])
        # stokesD = stokesD/stokesD[:,0]
        #
        # sys.stdout.writelines(str(stokesD))
        #
        # sopcV=self.calc_SOPC([output1[0].transpose(),stokes.transpose()])
        # sopcD=self.calc_SOPC([output1[0].transpose(),stokesD.transpose()])
        # sys.stdout.writelines(str(sopcV) + str(sopcD))
        # sys.stdout.flush()
        # if sopcV<sopcD:
        #    stokes=stokesD

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

        self.SET_VF_O(Low_Volt)
        sleep(0.05)
        SOP=stokes.transpose()
        self.control_pola(stokes[0,:].transpose(),thes=thres)
        self.SET_VF_CTL(0)
        return SOP,V

    def calc_SOPC(self,sop):
        Csop=sop
        reference=Csop[0]
        SOPC=[]
        for i in Csop[1:len(Csop)]:
            temp=reference[1:4].transpose()*i[1:4]
            SOPC.append(arccos(temp).item(0))
        return SOPC

    def control_pola(self,S,thes=0.009):
        '''

        exanple control_pola(matrix[[1],[0],[0],[1]])
        '''
        self.averageN=0
#        print "Jorge Castillo"
        #sys.stdout.flush()
        S1=[S,self.get_SOP_normalized()]
        e=self.calc_SOPC(S1)[0]
        i=0
        self.SET_POL_DRV1(2.5)
        self.SET_POL_DRV2(2.5)
        self.SET_POL_DRV3(2.5)
        self.SET_POL_DRV4(2.5)
        while float(e)>thes and (i<15 and float(e)>0.05)  and ~self.hwError:
            step, e =self.axis_optimizer(S,1)
            if e<thes:
                break
            step, e =self.axis_optimizer(S,2)
            if e<thes:
                break
            step, e =self.axis_optimizer(S,3)
            if e<thes:
                break
            step, e =self.axis_optimizer(S,4)
            #print e, float(e)
            i=i+1
            if i>20:
                self.hwError=True
                break
#        print 'Stop SOPC, ', e
        #sys.stdout.flush()
        return self.averageN,i
    def __setDRV__(self,voltage):
        if voltage < 0.1:
            self.fun_set(0.1)
            self.STATUS()
        elif voltage > 4.9:
            self.fun_set(4.9)
            self.STATUS()
        else:
            self.fun_set(voltage)
    def axis_optimizer(self,S,axis):
        delay=0.01
        step=0.2
        if axis==1:
            fun_get=self.GET_PC_MON1
            self.fun_set=self.SET_POL_DRV1
        if axis==2:
            fun_get=self.GET_PC_MON2
            self.fun_set=self.SET_POL_DRV2
        if axis==3:
            fun_get=self.GET_PC_MON3
            self.fun_set=self.SET_POL_DRV3
        if axis==4:
            fun_get=self.GET_PC_MON4
            self.fun_set=self.SET_POL_DRV4

        S1=[S,self.get_SOP_normalized()]
        #sys.stdout.flush()
        sopc=self.calc_SOPC(S1)
        if sopc[0]<0.1:
            step=step/4
        initial_set=fun_get()[0]+step
        self.__setDRV__(initial_set)
        sleep(delay)
        S1[1]=self.get_SOP_normalized()
        sopc.append(self.calc_SOPC(S1)[0])
        #print 'SOPC, ',sopc
        temp_drv=fun_get()[0]
        if abs(temp_drv - initial_set) >0.07 and temp_drv >=0.1 and temp_drv
<=4.9:
            self.hwError = True
            print 'the LLC '+self.__antennaName__+' '+str(axis)+' fail to set
the requested value: '
            sys.stdout.flush()
            return step,sopc[0]
        if (sopc[1]>sopc[0] or temp_drv>4.9) and temp_drv>0.1:
           step=-step
        i=0

        average=False
        t_start=time()
        while abs(sopc[i%2]-sopc[(i+1)%2])>(sopc[i%2]/40.0) and abs(sopc[i%2]-
sopc[(i+1)%2])>0.002 and (time()-t_start)<4:
           if (sopc[i%2]>sopc[(i+1)%2]):
               step=-step/2
           i=i+1
           temp_drv=temp_drv+step
#           print 'voltage, ', temp_drv
           #sys.stdout.flush()
           if temp_drv<0.1 or temp_drv>4.9:
               break
           self.__setDRV__(temp_drv)
           sleep(delay)
           S1[1]=self.get_SOP_normalized(average)
           sopc[i%2] = self.calc_SOPC(S1)[0]
           if sopc[i%2]<0.010:
               average=True
           #print 'SOPC, ',sopc[i%2]

        self.__setDRV__(temp_drv-step)
        return step, sopc[i%2]

    def calc_circle(self,s):

'''s.append([llc.GET_PC_MON1()[0],llc.GET_PC_MON2()[0],llc.GET_PC_MON3()[0],ll
c.GET_PC_MON4()[0],control_pola.get_SOP_normalized(llc.Calibration_matrix,llc)
])'''
        m1=[-s[0][4].item(2),-s[0][4].item(2),-s[0][4].item(3),1,-
s[0][4].item(1)**2-s[0][4].item(2)**2-s[0][4].item(3)**2+1]
        m2=[-s[1][4].item(2),-s[1][4].item(2),-s[1][4].item(3),1,-
s[1][4].item(1)**2-s[1][4].item(2)**2-s[1][4].item(3)**2+1]
        m3=[-s[2][4].item(2),-s[2][4].item(2),-s[2][4].item(3),1,-
s[2][4].item(1)**2-s[2][4].item(2)**2-s[2][4].item(3)**2+1]
        m4=[-s[3][4].item(2),-s[3][4].item(2),-s[3][4].item(3),1,-
s[3][4].item(1)**2-s[3][4].item(2)**2-s[3][4].item(3)**2+1]
        M=matrix([m1,m2,m3,m4])
        v=M[:,0:4].getI()*M[:,4]
        return v

    def calc_center(self,vectors):
        c=cross((vectors[0]-vectors[1]),(vectors[0]-vectors[3]))
        m=sqrt(vectors[0]*vectors[0].transpose)
        mc=sqrt(c*c.transpose)
        'c=c*m/mc'
        return c

    def find_center(self):
        vectors=[]
        self.SET_POL_DRV1(2.5)
        sleep(0.1)
        vectors.append(self.get_SOP_normalized())
        self.SET_POL_DRV1(2.7)
        sleep(0.1)
        vectors.append(self.get_SOP_normalized())
        self.SET_POL_DRV1(2.8)
        sleep(0.1)
        vectors.append(self.get_SOP_normalized())
        center=calc_center(vectors)
    def SOP_retriver(self,au):
        '''retriver 3 polarizing vectors
        the first one is the vector actualy used by the llc
        the second is the oposit vector
        the third one is the orthogonal vector of the last two
        '''

        #aux = [self.get_SOP_normalized(True)] #se obtiene el primer vector
desde el LLC
        aux=[au]
        aux1=matrix([aux[0][0,0],aux[0][1,0]*-1,aux[0][2,0]*-1,aux[0][3,0]*-
1]).transpose()
        aux.append(aux1) #el segundo vector se obtiene del vector opuesto al
anterior
        aux.append(ortogonal(aux)) # el tercer vector se obtiene utilizando el
metodo ortogonal
```

```
        return aux

    def __randomize_P__(self):
        import random
        delay=0.1

target=array([random.random()*4.0,random.random()*4.0,random.random()*4.0,rand
om.random()*4.0])+.1

current=array([self.GET_PC_MON1()[0],self.GET_PC_MON2()[0],self.GET_PC_MON3()[
0],self.GET_PC_MON4()[0]])
        delta=target-current

        current = current + delta*1.2
        current[current>5]=5
        current[current<0]=0

        try:
            #print pasox
            self.SET_POL_DRV1(current.item(0))
            sleep(delay)
            self.SET_POL_DRV2(current.item(1))
            sleep(delay)
            self.SET_POL_DRV3(current.item(2))
            sleep(delay)
            self.SET_POL_DRV4(current.item(3))
            sleep(delay)
        except :
            print 'Error setting POL_DRV'
        sleep(0.1)

        try:
            #print pasox
            self.SET_POL_DRV1(target.item(0))
            sleep(delay)
            self.SET_POL_DRV2(target.item(1))
            sleep(delay)
            self.SET_POL_DRV3(target.item(2))
            sleep(delay)
            self.SET_POL_DRV4(target.item(3))
            sleep(delay)
        except :
            print 'Error setting POL_DRV'
#                paso = paso/2.0
#                if any(abs(paso)>0.1):
#                    current = current + pasox
#                else:




def ortogonal(lista_vectores_encontrad):
    ''' method for calculate an orthonogal vector
    '''
    sop=copy(lista_vectores_encontrad[0])
    aux = [100,0]
    aux1=[]
    for i in range(1,4): #se busca el minimo elemento del vector polarizacion del LLC
        if numpy.minimum(aux[0],sop[i,0]) == sop[i,0]:
            aux = [sop[i,0],i]
    if aux[1] == 1:
        aux1=[0,sop[1,0],sop[2,0],(-(sop[1,0]**2+sop[2,0]**2)/sop[3,0])]
    elif aux[1]==2:
        aux1=[0,sop[1,0],(-(sop[1,0]**2+sop[3,0]**2)/sop[2,0]),sop[3,0]]
    else:
        aux1=[0,(-(sop[2,0]**2+sop[3,0]**2)/sop[1,0]),sop[2,0],sop[3,0]]
    aux1=matrix(aux1)
    aux1[0,0]=(numpy.sqrt(aux1*aux1.transpose())) #se calcula norma del vector
    for i in range(1,4):
        aux1[0,i]=aux1[0,i]/aux1[0,0]
    return aux1.transpose()


def list_to_matrix(lista1):
    '''Generates a matrix with both SOP vectors obtained for each polarization with the stretcher  at 0V and 5V
    the output element is formated for the method list_to_matrix1
    '''

    aux=zeros((size(lista1[0]),len(lista1)))
    for i in range(0, size(aux[:,0])):
        for h in range(0, size(aux[0,:])):
            aux[i,h]=lista1[h][0,i]
    return aux

def list_to_matrix1(lista1):
    '''Generates a matrix with the elements within the list, in this case the list will be the output matrices of each polarization state
    The matrix is formated for using it with the Stokes2Jones method included in StokesJones.py
    '''

    aux=zeros((len(lista1)*len(lista1[0][0,:]),size(lista1[0][:,0])))
    for i in range(0, size(aux[:,0])):
        if i==0 or i==1:
            for h in range(0,4):
                aux[i,h]=lista1[0][h,i]
        if i==2 or i==3:
            for h in range(0,4):
                aux[i,h]=lista1[1][h,i-2]
        if i==4 or i==5:
            for h in range(0,4):
                aux[i,h]=lista1[2][h,i-4]
    return matrix(aux)
```

StokesJones.py:

This script implements conversion between Jones and Stokes space:

```
from scipy import *
from numpy import *
import sys,numpy, numbers

def Jones2Stokes(J):
    '''Converts Jones matrix into Stokes matrix
    the input must be a n x 4 matrix
    '''
    #print J
    sys.stdout.flush()
    temp1=abs(J.item(0))**2
    temp2=abs(J.item(1))**2
    S1=temp1+temp2
    S2=temp1-temp2
    S3=2*((abs(J.item(0)+J.item(1))/2)**2-(abs(J.item(0)-J.item(1))/2)**2)
    temp1=1j*J.item(1)
    S4=2*((abs(J.item(0)+temp1)/2)**2-(abs(J.item(0)-temp1)/2)**2)
    return matrix([S1,S2,S3,S4])

def stokes(S1,S2,S3,S4):
    s=zeros((numpy.size(S1[:,0]),4))
    for n in range(0,4):
        for m in range(numpy.size(S1[:,0])):
            if n==0:
                s[m,n]=S1[m,0]
            if n==1:
                s[m,n]=S2[m,0]
            if n==2:
                s[m,n]=S3[m,0]
            if n==3:
                s[m,n]=S4[m,0]
    return s

def veclen(matriz):
    '''
    estimates the square root of the sum for each row of the givn matrix
    '''

    x = matriz.copy()
    x = x.transpose()
    if numpy.size(x)==0:
        return
    else:
        y=matrix(numpy.zeros((numpy.size(x[0]),(numpy.size(x[1]))/3)))
        idx=0
        for k in range(0,numpy.size(x[1])):
            y[k,0]=sqrt(suma(x[:,k]))
            idx=idx+1
        return y.transpose()

def suma (x):
    '''calculates the sum of a row of the given matrix
    the input variable must have the format a[:,row]
    '''

    y=0
    for i in range(0,numpy.size(x)):
        y+=x[i,0]*x[i,0]
    return y

def Stokes2Jones(S):
    '''Converts Jones matrices into Stokes matrices
    the input must be a nx2 matrix
    '''

    s=S[:,1:4]
    l=veclen(s);
    l=l.transpose()
    l=fill_matrix(l,3)
    s=s/l
    j1=sqrt(0.5*(s[:,0]+1))
    j2=sqrt(-0.5*(s[:,0]-1))
    phi=arctan2(0.5*s[:,2],0.5*s[:,1])/2
    aux1=[]
    aux2=[]
    for i in range(0,(numpy.size(j1))):
        aux1.append(j1[i,0]*numpy.exp(complex(0,phi[i,0])))
        aux2.append(j2[i,0]*numpy.exp(complex(0,(-phi[i,0]))))
    aux1=matrix(aux1).transpose()
    aux2=matrix(aux2).transpose()
    return scalar_multiply([aux1, aux2],sqrt([S[:,0], S[:,0]]))

def fill_matrix(vector, times):
    '''generates a matrix with n times the same vector
    '''

    aux= numpy.zeros((numpy.size(vector[:,0]), times))
    for n in range(0,numpy.size(vector[:,0])):
        fill(aux,vector,n,times)
    return aux

def fill(zeros,vector,n,times):
    if times<0:
        return
    else:
        zeros[n,(times-1)]=vector[n,0]
        fill(zeros,vector,n,times-1)


def scalar_multiply(lista1, lista2):
    '''calculates the point to point multiplication between two matrices
    both matrices must have the saame dimensions
    '''

    aux1=zeros((numpy.size(lista1[0]),len(lista1)),complex)
    aux2=zeros((numpy.size(lista2[0]),len(lista2)),complex)
    for i in range(0, numpy.size(aux1[:,0])):
        for h in range(0, numpy.size(aux1[0,:])):
            aux1[i,h]=lista1[h][i,0]
            aux2[i,h]=lista2[h][i,0]
    return matrix(multiply(aux1,aux2))
```

transmission.py:

This script implements the construction of the Jones transmission matrix.

```
from numpy import *
import sys



def k1(hx,hy):
    return hx/hy

def k2(vx,vy):
    return vx/vy

def k3(qx,qy):
    return qx/qy

def k4 (k1,k2,k3):
    return (k3-k2)/(k1-k3)


def Transmission_matrix(h,v,q):
    '''this method allows the user to calculate the transmission matrix of a
electromagnetic wave
    the vectors must be matrices containing a Jones vector
    '''

    r1=k1(h[0,0], h[0,1])
    r2=k2(v[0,0], v[0,1])
    r3=k3(q[0,0], q[0,1])
    r4=k4(r1,r2,r3)
    aux = matrix(([[r1*r4,r2],[r4,1]]),complex)

    return aux
```

doLlcPolarizationCalibration.py:

This is the user script used to start a concurrent optimization of many LLCs at once

```
#! /usr/bin/env python
#
# $Id: doLlcPolarizatioCalibration.py,v 1.7 2013/01/28 12:18:20 jcastill Exp $
#
# ALMA - Atacama Large Millimiter Array
# (c) Associated Universities Inc., 2006
#
# This library is free software; you can redistribute it and/or
# modify it under the terms of the GNU Lesser General Public
# License as published by the Free Software Foundation; either
# version 2.1 of the License, or (at your option) any later version.
#
# This library is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# Lesser General Public License for more details.
#

__author = "$Author: jcastill $"
__date__ = "$Date: 2013/01/28 12:18:20 $"
__version__ = "$Revision: 1.7 $"

import sys
from optparse import OptionParser
from subprocess import Popen
import sys
from time import sleep

from JLLC import JLLC
from CCL.SAS import SAS
import lib.BEDefinitions as Bdef



def doPolarizationCalibration(Antenna_name):

#============================================================================
    # print Bdef.YELLOW + "please optimize the SAS PBS power before running
the calibration for " + Antenna_name +Bdef.NO_COLOR
    # sys.stdout.flush()
    # return

#============================================================================
    try:
        llc=JLLC(Antenna_name)
        sas = SAS(Antenna_name)
    except:
        print Bdef.RED + 'the ' + Antenna_name + ' LLC is not available'
+Bdef.NO_COLOR
        return
    if sas.GET_PBS_OPT_DET()[0]<-1:
        print Bdef.YELLOW + "please optimize the SAS PBS power before running
the calibration for " + Antenna_name +Bdef.NO_COLOR
        sys.stdout.flush()
        return
    SOP=llc.pola_optimization(0.2, 4.5)

#===========================================================================
    # llc.SET_CAL_CHECK(1)
    # while llc.GET_ROUTINE_STATUS()[0][0]==1:
    #    i=1
    # CAL = llc.GET_CAL_RESULT()[0][0]
    # SOPC = llc.GET_SOPC()[0]
    # print 'Antenna ',Antenna_name,' final SOPC :',SOPC
    # sys.stdout.flush()

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


def main():
    parser = OptionParser()
    parser.set_usage("""
    doLlcPolarizatioCalibration.py [antenna list] [options]

    optimize the polarization of one or more antennas's LLCs.
    examples:

    doLlcPolarizatioCalibration.py CM02 CM01 CM05
    doLlcPolarizatioCalibration.py -a Array001
    doLlcPolarizatioCalibration.py

    run the script with no arguments to optimize all the antennas in the STE.
    """)
    parser.add_option("-a", "--array", action="store", type="string",
dest="array",
                      metavar="ARRAY_NAME",help="process antennas on given
array.  Example: -a Array002")
    (options, args) = parser.parse_args()


    # Antenna selection

    if options.array is not None:
        import GetAntennas
        try:
            import GetAntennas
            ga = GetAntennas.GetAntennas(options.array,True)
        except:
            print 'Array \'%s\' does not exist' % options.array
            return
        sAntennaList = ga.sAntennaList

    if len(args)>1:
        sAntennaList = args
    elif 1 == len(args):

#=======================================================================
        # if a single antenna is provided just start the optimization
```

```
#=======================================================================
        doPolarizationCalibration(args[0])
        return
    elif options.array is None:
        import GetAntennas
        ga = GetAntennas.GetAntennas()
        sAntennaList = ga.getAvailableAntennas()

#============================================================================
    # end antenna selection

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

    inPut=''
    while (not ( inPut in( ['y','n']))):
        print '\n\n The LLC of the antennas ' , sAntennaList , ' will be
optimizedn do you want to continue? y/n: '
        inPut=raw_input()
    if inPut is 'n':
        return
    sys.stdout.flush()

#============================================================================
    # start the parallel optimization process

#============================================================================
    p=[]
    isChecked = []
    for i in sAntennaList:
        p.append(Popen(["doLlcPolarizatioCalibration.py", i]))
        isChecked.append(False)

    n=len(p)

    salerito='\\'


    while(n>0):
        if salerito == '\\':
            salerito = '|'
        elif salerito == '|':
            salerito = '/'
        elif salerito == '/':
            salerito = '-'
        elif salerito == '-':
            salerito = '\\'
        sys.stdout.writelines('not optimized LLCs # '+ n.__str__()+' ' +
salerito +'\r')
        sys.stdout.flush()
        n=0
        sleep(0.2)
        outs = []
        for idx, i in enumerate (p):
            #outs.append(i.poll())

            if (i.poll() is None):
                n=n+1
            elif not isChecked[idx]:
                llc = JLLC(sAntennaList[idx])
                llc.SET_CAL_CHECK(1) #initialize the calibration check routine
on the LLC and continue
                del llc
                isChecked[idx] = True # set the antenna status to checked
(True) to avoid to recheck
    fail_antennas = []

#===========================================================================
    # for idx,i in enumerate(outs):
    #    if i == 1:
    #        fail_antennas.append(sAntennaList[idx])

#===========================================================================
    print 'Optimization of requested LLCs is finished\n'

#=============================================================================
==
# get the calibrations results for all the antennas
#=============================================================================
==
    for i in sAntennaList:
        try:
            llc=JLLC(i)
        except:
            print 'the ',i,' LLC is not available'
            continue
        while llc.GET_ROUTINE_STATUS()[0][0]==1:
            pass
        sopc = llc.GET_SOPC()[0]
        del llc
        print 'Antenna ',i,' final SOPC :', sopc
        if sopc > 0.1 :
            fail_antennas.append(i)
    if len(fail_antennas)>0:
        sys.stdout.writelines (Bdef.GREEN + 'the antennas: ' + Bdef.NO_COLOR)
        for i in fail_antennas:
            sys.stdout.writelines(Bdef.GREEN + i + ' ' + Bdef.NO_COLOR)
        print Bdef.GREEN + 'are not correctly optimized please run the script
again for them' +Bdef.NO_COLOR + '\n'



#===========================================================================
    # Main loop

#===========================================================================
if __name__ == "__main__":

    main()

    sys.stdout=open('/dev/null','w')
```

Test.py:

This script was used to test the algorithm, on many LLC at the same time

```
import random,sys
import thread, threading,time
from JLLC import JLLC
from pylab import matrix,cross,arccos,pi
from numpy import *
from time import sleep,time
import csv
from speedTest import *
from CCL.SAS import SAS

def randomSOP():
    a=[]
    for i in range(4):
        a.append(random.random())
    a[0]=sqrt(a[1]**2+a[2]**2+a[3]**2)
    a=matrix(a)
    a=a/a[0,0]
    return matrix(a.transpose())

def Tests(points):
    file = open('testSOP.txt','w')
    file.write('SOPesperado,SOPobtenido,SOPC'+'\n')
    for i in range(points):
        SOP=randomSOP()
        llc=JLLC('DA60',stickyFlag=True)
        llc.control_pola(SOP)
        SOP_NEW = llc.get_SOP_normalized()
        SOPC = llc.calc_SOPC([SOP,SOP_NEW])

file.write(str(SOP.transpose())+','+str(SOP_NEW.transpose())+','+str(SOPC[0])+
'\n')
    return

def repeat(points):
    file=open('repeat.txt','w')
    file.write('SOPesperado,SOPobtenido,SOPC'+'\n')
    SOP = randomSOP()
    llc=JLLC('DA60',stickyFlag=True)
    for i in range(points):
        llc.control_pola(SOP)
        SOP_NEW = llc.get_SOP_normalized()
        SOPC = llc.calc_SOPC([SOP,SOP_NEW])

file.write(str(SOP.transpose())+','+str(SOP_NEW.transpose())+','+str(SOPC[0])+
'\n')
    return
def Test(antenna,points,low,high,no_spread=False):
    '''This method is made for statistical purpose. It can be used either in a
single antenna or in an array.
    For single antenna it is needed the name of the antenna and the number of
points to take, for example: Test('DV08',10)
```

```
    For an array it is needed a list of antennas, for example:
Test(['DV08','DV18','DA60'],10)
    '''

    n=points
    if isinstance(antenna,str):
        interval = float(((high-low)/2.0)/n)
        if no_spread:
            interval=0
        #turn_on(l)
        llc=JLLC(antenna,stickyFlag=True)
        turn_on(llc)
        file = open ('test_for_the_LLC_of_the_antenna_N_' + antenna
+'.txt','w')
        fil = open ('test_for_antenna_N_' + antenna +'.txt','w')
        file.write('Low Voltage,High
Voltage,Delta,Time,SOPReal,SOPDesierd,Calibration Status,SOPCsys,Eigenvector'
+ '\n')
        fil.write('Low Voltage,High Voltage,Delta,Time,Calibration
Status,SOPCsys,SOPCdif,iS0,iS1,iS2,iS3' + '\n')
        for i in range (n):
            llc.__randomize_P__()
            print i, antenna
            sopI=llc.get_SOP_normalized()

            Low_Voltage=  low+(i*interval)
            High_Voltage= high-(i*interval)
            delta=High_Voltage-Low_Voltage
            if Low_Voltage != High_Voltage:
                start = time()
                SOP=llc.pola_optimization(Low_Voltage, High_Voltage)
                end= (time()-start)
                SOP1=llc.get_SOP_normalized()
                SOPC1= llc.calc_SOPC([SOP[0],SOP1])[0]
                llc.SET_CAL_CHECK(1)
                while llc.GET_ROUTINE_STATUS()[0][0]==1:
                    i=1
                CAL = llc.GET_CAL_RESULT()[0][0]
                SOPC = llc.GET_SOPC()[0]

file.write(str(Low_Voltage)+','+str(High_Voltage)+','+str(delta)+','+str(end)+
','+str(SOP1.transpose())+','+str(SOP[0].transpose())+','+str(CAL)+','+str(SOP
C)+','+str(SOPC1)+','+str(SOP[1].transpose())+'\n')

fil.write(str(Low_Voltage)+','+str(High_Voltage)+','+str(delta)+','+str(end)+'
,'+str(CAL)+','+str(SOPC)+','+str(sopI[0])+','+str(sopI[1])+','+str(sopI[2])+'
,'+str(sopI[1])+'\n')
                file.flush()
                fil.flush()
        file.close()
        fil.close()
    elif isinstance(antenna,list):
        t=[]
        for i in range(len(antenna)):
            t.append(TestThread(antenna[i],n,low,high,no_spread))

        for i in range(len(antenna)):
            t[i].start()
        for i in range(len(antenna)):
            t[i].join()
        consolida()

def TestC(antenna,points,low,high,no_spread=False):
    '''This method is made for statistical purpose. It can be used either in a 
single antenna or in an array.
    For single antenna it is needed the name of the antenna and the number of 
points to take, for example: Test('DV08',10)
    For an array it is needed a list of antennas, for example: 
Test(['DV08','DV18','DA60'],10)
    '''
    print 'prueba con reintento'
    sys.stdout.flush()
    n=points
    if isinstance(antenna,str):
        interval = float(((high-low)/2.0)/n)
        if no_spread:
            interval=0
        #turn_on(l)
        llc=JLLC(antenna,stickyFlag=True)
        sas=SAS(antenna,stickyFlag=True)
        turn_on(sas)
        turn_on(llc)
        file = open ('test_op_alg_e_r_' + antenna +'.txt','w')
        fil = open ('test_op_alg_r_' + antenna +'.txt','w')
        file.write('Low Voltage,High 
Voltage,Delta,Time,SOPReal,SOPDesierd,Calibration Status,SOPCsys,Eigenvector' 
+ '\n')
        fil.write('Low Voltage,High Voltage,Delta,Time,Calibration 
Status,SOPCsys,N_try,iS1,iS2,iS3,r_try,sas_switch_Port,sas_pbs,n_antenna,Anten
na' + '\n')
        for i in range (n):
            counter=1
            llc.__randomize_P__()
            print i, antenna
            sys.stdout.flush()
            Low_Voltage=  low+(i*interval)
            High_Voltage= high-(i*interval)
            delta=High_Voltage-Low_Voltage
            sopI=llc.get_SOP_normalized()
            if Low_Voltage != High_Voltage:
                start = time()
                SOP=llc.pola_optimization(Low_Voltage, High_Voltage)
                end= (time()-start)
                SOP1=llc.get_SOP_normalized()
                SOPC1= llc.calc_SOPC([SOP[0],SOP1])[0]
                llc.SET_CAL_CHECK(1)
                while llc.GET_ROUTINE_STATUS()[0][0]==1:
                    pass
                CAL = llc.GET_CAL_RESULT()[0][0]
                SOPC=llc.GET_SOPC()[0]
                while CAL==0 or SOPC>0.11:
                    
fil.write(str(Low_Voltage)+','+str(High_Voltage)+','+str(delta)+','+str(end)+'
,'+str(CAL)+','+str(SOPC)+','+ str(counter)
+','+str(sopI.item(1))+','+str(sopI.item(2))+','+str(sopI.item(3))+',1,'+str(s
as.GET_SWITCH_PORT()[0])+','+str(sas.GET_PBS_OPT_DET()[0])+'\n')
                    fil.flush()
                    SOP=llc.pola_optimization(Low_Voltage, High_Voltage)
                    print 'reintentando'
                    llc.SET_CAL_CHECK(1)
                    while llc.GET_ROUTINE_STATUS()[0][0]==1:
                        sleep(0.1)
                    CAL = llc.GET_CAL_RESULT()[0][0]
                    SOPC=llc.GET_SOPC()[0]
                    counter = counter + 1
                end= (time()-start)

file.write(str(Low_Voltage)+','+str(High_Voltage)+','+str(delta)+','+str(end)+
','+str(SOP1.transpose())+','+str(SOP[0].transpose())+','+str(CAL)+','+str(SOP
C)+','+str(SOPC1)+','+str(SOP[1].transpose())+'\n')

fil.write(str(Low_Voltage)+','+str(High_Voltage)+','+str(delta)+','+str(end)+'
,'+str(CAL)+','+str(SOPC)+','+ str(counter)
+','+str(sopI.item(1))+','+str(sopI.item(2))+','+str(sopI.item(3))+',0,'+str(s
as.GET_SWITCH_PORT()[0])+','+str(sas.GET_PBS_OPT_DET()[0])+'\n')
                file.flush()
                fil.flush()
        file.close()
        fil.close()
    elif isinstance(antenna,list):
        t=[]
        for i in range(len(antenna)):
            t.append(TestThreadC(antenna[i],n,low,high,no_spread))

        for i in range(len(antenna)):
            t[i].start()
        for i in range(len(antenna)):
            t[i].join()
        consolida_test_op_alg()

def TestB(antenna,points):
    '''This method is made for statistical purpose. It can be used either in a
single antenna or in an array.
    For single antenna it is needed the name of the antenna and the number of
points to take, for example: Test('DV08',10)
    For an array it is needed a list of antennas, for example:
Test(['DV08','DV18','DA60'],10)
    '''

    n=points
    if isinstance(antenna,str):
        interval=0
        low=0
        high=5
        #turn_on(l)
        llc=JLLC(antenna,stickyFlag=True)
        turn_on(llc)
        file = open ('test_e_BL_' + antenna +'.txt','w')
        fil = open ('test_BL_' + antenna +'.txt','w')
        file.write('Time,SOPReal,SOPDesierd,Calibration
Status,SOPCsys,Eigenvector\n')
```

```
        fil.write('Time,Calibration
Status,SOPCsys,iS0,iS1,iS2,iS3,n_antenna,antenna\n')
        for i in range (n):
            llc.__randomize_P__()
            print i, antenna
            sopI=llc.get_SOP_normalized()
            Low_Voltage=  low+(i*interval)
            High_Voltage= high-(i*interval)
            delta=High_Voltage-Low_Voltage
            if Low_Voltage != High_Voltage:
                start = time()
                llc.SET_CAL_START(1)
                SOP=[0]
                while llc.GET_ROUTINE_STATUS()[0][0]==1:
                    sleep(0.1)
                end= (time()-start)
                SOP1=llc.get_SOP_normalized()
                SOPC1= 0
                llc.SET_CAL_CHECK(1)
                while llc.GET_ROUTINE_STATUS()[0][0]==1:
                    sleep(0.1)
                CAL = llc.GET_CAL_RESULT()[0][0]
                SOPC = llc.GET_SOPC()[0]

file.write(str(end)+','+str(SOP1.transpose())+','+str(SOP[0])+','+str(CAL)+','
+str(SOPC)+','+str(SOPC1)+','+str(SOP[0])+'\n')

fil.write(str(end)+','+str(CAL)+','+str(SOPC)+','+str(sopI.item(0))+','+str(so
pI.item(1))+','+str(sopI.item(2))+','+str(sopI.item(1))+'\n')
                file.flush()
                fil.flush()
        file.close()
        fil.close()
    elif isinstance(antenna,list):
        t=[]
        for i in antenna:
            t.append(TestThreadB(i,n))

        for i in t:
            i.start()
        for i in t:
            i.join()
        consolida()

def Test_de_control(antenna,points):
    '''This method is made for statistical purpose. It can be used either in a
single antenna or in an array.
    For single antenna it is needed the name of the antenna and the number of
points to take, for example: Test('DV08',10)
    For an array it is needed a list of antennas, for example:
Test(['DV08','DV18','DA60'],10)
    '''

    n=points
    if isinstance(antenna,str):

        llc=JLLC(antenna,stickyFlag=True)
        turn_on(llc)
        speedTest(n,antenna,llc)
    elif isinstance(antenna,list):
        t=[]
        for i in antenna:
            t.append(TestThread_de_control(i,n))
        for i in t:
            i.start()
        for i in t:
            i.join()
        speedTestControl()

class TestThread(threading.Thread):
    def __init__(self,antenna,points,low,high,no_spread=False):
        threading.Thread.__init__(self)
        self.antenna=antenna
        self.points = points
        self.low=low
        self.high=high
        self.no_spread=no_spread

    def run(self):
        Test(self.antenna,self.points,self.low,self.high,self.no_spread)

class TestThreadC(threading.Thread):
    def __init__(self,antenna,points,low,high,no_spread=False):
        threading.Thread.__init__(self)
        self.antenna=antenna
        self.points = points
        self.low=low
        self.high=high
        self.no_spread=no_spread

    def run(self):
        TestC(self.antenna,self.points,self.low,self.high,self.no_spread)

class TestThreadB(threading.Thread):
    def __init__(self,antenna,points):
        threading.Thread.__init__(self)
        self.antenna=antenna
        self.points = points

    def run(self):
        TestB(self.antenna,self.points)
class TestThread_de_control(threading.Thread):

    def __init__(self,antenna,points):
        threading.Thread.__init__(self)
        self.antenna=antenna
        self.points = points

    def run(self):
        Test_de_control(self.antenna,self.points)

def probe(antenna,low,delta_mayor,delta_menor,n):
    high=low+delta_mayor
    step = float((delta_mayor-delta_menor)/n)
    llc=JLLC(antenna,stickyFlag=True)
    file = open ('test_for_the _LLC of_the_antenna_' + antenna +'.txt','w')

file.write('Low_Voltage,High_Voltage,Delta,Time,Calibration_Status,SOPCsys,SOP
Cdif' + '\n')
    i=0
    delta=delta_mayor
    Low_Voltage=  low
    High_Voltage= high
    while delta>=delta_menor:
        print i
        if Low_Voltage != High_Voltage:
            start = time.time()
            SOP=llc.pola_optimization(Low_Voltage, High_Voltage)
            end= (time.time()-start)
            SOP1=llc.get_SOP_normalized()
            SOPC1= llc.calc_SOPC([SOP,SOP1])[0]
            llc.SET_CAL_CHECK(1)
            while llc.GET_ROUTINE_STATUS()[0][0]==1:
                h=1
            CAL = llc.GET_CAL_RESULT()[0][0]
            SOPC = llc.GET_SOPC()[0]

file.write(str(Low_Voltage)+','+str(High_Voltage)+','+str(delta)+','+str(end)+
','+str(CAL)+','+str(SOPC)+','+str(SOPC1)+'\n')
        i+=1
        Low_Voltage=  low+(i*step)
        High_Voltage= high-(i*step)
        delta=High_Voltage-Low_Voltage
    file.close()

def pola_obt(antenna):
    if isinstance(antenna,str):
        llc=JLLC(antenna)
        llc.pola_optimization(2,3)#agregar voltajes fijos para polarizar
    elif isinstance(antenna,list):
        t=[]
        for i in range(len(antenna)):
            t.append(Thread(antenna[i]))
        for i in range(len(antenna)):
            t[i].start()
        for i in range(len(antenna)):
            t[i].join()

class Thread(threading.Thread):
    def __init__(self,antenna):
        threading.Thread.__init__(self)
        self.antenna=antenna

    def run(self):
        pola_obt(self.antenna)

def turn_on(dev):
    '''
    Turns on the device by going through the HW device lifecycle,
    leaving it in Operational state. Assumes the device currently
    being in the Start state.
    Example:
    dgck = DGCK("DV01")
    turnon(dgck)
    '''
    import ControlDeviceExceptions
    state = '0'
    for key, val in dev._devices.iteritems():
        print "Device state was:", dev._devices[key].getHwState()
        print "Going through HW lifecycle for %s" % key
        try:
            if str(dev._devices[key].getHwState())=="Stop":
                dev._devices[key].hwStart()
                print "Device state changed to  %s 
successfully"%dev._devices[key].getHwState()
                state = '1'
            state = '1'
            if str(dev._devices[key].getHwState())=="Start":
                dev._devices[key].hwConfigure()
                print "Device state changed to  %s 
successfully"%dev._devices[key].getHwState()
                state = '2'
            state = '2'
            if str(dev._devices[key].getHwState())=="Configure":
                dev._devices[key].hwInitialize()
                print "Device state changed to  %s 
successfully"%dev._devices[key].getHwState()
                state= '3'
            state = '3'
            if str(dev._devices[key].getHwState())=="Initialize":
                dev._devices[key].hwOperational()
                print "Device state changed to  %s 
successfully"%dev._devices[key].getHwState()
                state = '4'
            state = '4'
        except ControlDeviceExceptions.HwLifecycleEx, ex:
            print "Error changing state of device!!!"
            #__ex_pretty_printer(str(ex.errorTrace))
            if (state == '1'):
                print "Device %s could not go to Configure State. Likely 
device is not responding to SN CanBus request."%key
            if (state == '2'):
                print "Device %s could not go to Initialize State."% key
            if (state == '3'):
                print "Device %s could not go to Operational State."% key
        if (state == '4'):
            print ">>> turn_on succeeded for %s " % key
        print "\n"
        print "Device state is now:", dev._devices[key].getHwState()
    return
def consolida_test_op_alg():
    import os
    import csv
    import pylab
    l=os.listdir(os.getcwd())
    l.sort()
    fo=open('test_op_alg.csv','w')
    csvfo=csv.writer(fo)
    m=0
    for j,i in enumerate(l):
        if i[0:14]=='test_op_alg_r_':

            fi=open(i,'r')
            cvsfi=csv.reader(fi)
            linea=cvsfi.next()
            if m==0:
                csvfo.writerow(linea)
            for k in cvsfi:
                k.append(m+1)
                k.append(i[14:18])
                csvfo.writerow(k)
            fi.close()
            m=m+1
    fo.close()
    data=pylab.csv2rec('test_op_alg.csv')
    pylab.subplot(121)
    pylab.scatter(data['time'],data['sopcsys'],c=data['n_antenna'])
    pylab.subplot(122)
    pylab.hist(data['time'], 30)
    pylab.show()

def consolida_speedTest():
    import os
    import csv
    import pylab
    l=os.listdir(os.getcwd())
    l.sort()
    fo=open('speedTest_C.csv','w')
    csvfo=csv.writer(fo)
    m=0
    for j,i in enumerate(l):
        if i[0:16]=='speedTestControl':
            fi=open(i,'r')
            cvsfi=csv.reader(fi)
            linea=cvsfi.next()
            if j==0:
                csvfo.writerow(linea)
            for k in cvsfi:
                k.append(m+1)
                k.append(i[21:25])
                csvfo.writerow(k)
            fi.close()
            m=m+1
    fo.close()
    data=pylab.csv2rec('speedTest_C.csv')
    pylab.hist(data['T'],30)
    pylab.show()

def consolida_BL():
    import os
    import csv
    import pylab
    l=os.listdir(os.getcwd())
    l.sort()
    fo=open('test_C_BL.csv','w')
    csvfo=csv.writer(fo)
    m=0
    for j,i in enumerate(l):
        sys.stdin.flush()
        if i[0:8]=='test_BL_':
            fi=open(i,'r')
            cvsfi=csv.reader(fi)
            linea=cvsfi.next()
            if m==0:
                csvfo.writerow(linea)
            for k in cvsfi:
                k.append(m+1)
                k.append(i[8:12])
                csvfo.writerow(k)
            fi.close()
            m=m+1
    fo.close()
    data=pylab.csv2rec('test_C_BL.csv')
    pylab.subplot(121)
    pylab.scatter(data['time'],data['sopcsys'],c=data['n_antenna'])
    pylab.subplot(122)
    pylab.hist(data['time'], 30)
    pylab.show()

def checksas(lista):
    l=[]
    for i in lista:
        sas=SAS(i,stickyFlag=True)
        turn_on(sas)
        if sas.GET_SWITCH_PORT()[0]>4:
            sas.SET_SWITCH_PORT(1)
        if sas.GET_PBS_OPT_DET()[0]<0:
            l.append(i)
        print i,sas.GET_SWITCH_PORT()[0],sas.GET_PBS_OPT_DET()[0]
    return l

def optim_sas(lista):
    for i in lista:
        sas=SAS(i,stickyFlag=True)
        turn_on(sas)
        sas.SET_OPTIMIZE_POL1(1)
    for i in lista:
        sas=SAS(i,stickyFlag=True)
        turn_on(sas)
        while sas.GET_ROUTINE_STATUS_PBS():
            sleep(0.1)
```

Author List of Publications

Article:

- J. A. Castillo and E. A. Michael, “Fast Algorithm to Find the Minimal Polarization Change Points on a Fiber-Optic Stretcher Using One Polarization Rotator and a Polarimeter,” *IEEE Photonics Journal*, vol. 4, no. 6, pp. 2390–2397, Dec. 2012.

Conference:

- J. Castillo, C. Barrientos, V. H. Calle, L. Pallanca, F. P. Mena, and E. A. Michael, “A New Laboratory for Terahertz Photonics,” in *20th International symposium on Space Terahertz Technology*, 2009, pp. 239–241.

- J. A. Castillo and E. A. Michael, “Practical implementation and on the field test results of an enhanced algorithm for polarization optimization of the line length corrector of ALMA central local oscillator,” in *SPIE 8452*, 2012, p. 84522N–84522N.