

Compendio de investigaciones en Geofísica

Trabajos de Investigación realizados por estudiantes **durante los años 2018 - 2019**



PERÚ

Ministerio del Ambiente



EL PERÚ PRIMERO

CRÉDITOS

Título: Compendio de investigaciones en Geofísica

Título clave abreviado: Compend. investig Geof.

Volumen: 19

ISSN: 2079-696X

Ministra del Ambiente

Fabiola Muñoz

Presidente Ejecutivo del IGP

Hernando Tavera

Director Científico del IGP

Danny Scipión

Editores:

Danny Scipión

Juan Carlos Gómez

Yamina Silva

Diseño y diagramación:

Luis Miguel Ybañez

Editado por:

Instituto Geofísico del Perú

Calle Badajoz 169, Urb. Mayorazgo, IV Etapa, Ate Vitarte

Central telefónica: (+ 51 1) 317 2300

www.gob.pe/igp

Lima, junio de 2020

PRÓLOGO	5
DE LOS EDITORES	6
SECCIÓN 1. Ciencias del geoespacio y astronomía	7
Caracterización de la capa límite atmosférica en el valle del Mantaro empleando radiosondas y sensores remotos <i>Josep Josué Prado Fasanando</i>	8
Conversión de programas en matlab de V.Paznukov a python para escalado automático de ionogramas <i>Luis Daniel Cárdenas Andrade</i>	13
Desarrollo de un sistema de alarma inteligente para el monitoreo de experimentos con el radar principal <i>Nicol Fiorella Huamán Molina</i>	18
Desarrollo de una interfaz software basada en GNU radio para controlar el sistema de adquisición JARS 2.0 del Radio Observatorio de Jicamarca <i>Isaac Mario Tupac Dávila</i>	23
Detección de ecos ionosféricos en ionogramas digitales utilizando redes neuronales convolucionales <i>César de la Jara Sánchez</i>	27
Diseño de un sistema de detección temprana de huaicos utilizando procesamiento de imágenes <i>Juan Carlos Barbaran Meza</i>	32
Diseño de una arquitectura de software para procesamiento paralelo escalable en signal chain <i>Sergio Cortez</i>	37
Diseño de una interfaz de control y supervisión para transmisores ATRAD <i>George Steve Fajardo Soria</i>	42
Diseño e implementación de un sistema de radar Ionosonda usando un hardware de código abierto <i>Francisco Sulca Jota</i>	46
Diseño e implementación de un sistema digital para la medición del desfase entre dos señales <i>Isaac Mario Tupac Dávila</i>	50
Diseño e implementación de un sistema electro-mecánico de radar de apertura sintética con control de apunte para modos stripmap, spotlight y scan <i>Camilo Daniel Ortecho Huamanchumo</i>	54
Envío de datos por gigabit ethernet desde una interfaz lvds usando soc (single board computer + fpga) <i>David George Shatwell Pittaluga</i>	59
Estudio de efectos ionosféricos en las señales de radiofrecuencia mediante el análisis de datos de receptores gnss <i>George Steve Fajardo Soria</i>	64
Estudio de las capas E esporádicas de tipo blanketing sobre el Radio Observatorio de Jicamarca <i>José Manuel Suclupe Osorio</i>	68
Generador de pulsos sincronizados con GPS usando api rest <i>Juan Marcos Luna Jaramillo</i>	73
Grabado de datos en formato digital RF signal para JARS 1.1 <i>Daniel Carrión Huacani</i>	78
Implementación de rutinas para la estimación de vientos y precipitación de un radar perfilador de vientos de antena espaciada que opera en UHF <i>Erick Bocanegra Salas</i>	83
Implementación de sistema de radar ionosonda en red pitaya <i>Alejandro Jesús Castro Ortecho</i>	88
Implementación y análisis de algoritmos de formación de imágenes para radares de apertura sintética de estación terrena	93

(gb-sar) de monitoreo de deslizamientos <i>Luis De la Cruz Painado</i>	98
Integración del radar BLTR con las aplicaciones realtime y la base de datos madrigal <i>Geordano Wilfredo Chagua Hidalgo</i>	102
Métodos Inversos en tomografía de la Ionósfera Ecuatorial <i>Juan Pablo Velásquez Ormaeche</i>	107
Monitoreo del espectro de potencia para detección de beacons satelitales <i>Jhoseling Milagros Melgarejo Figueroa</i>	111
Optimización del telescopio Nishimura del Instituto Geofísico del Perú para su uso científico en astronomía <i>Mario Zegarra-Valles</i>	116
Procesamiento y puesta en línea de Los datos de Radar Bltr/Huancayo <i>Jenny Margot Ramos Lázaro</i>	121
Radio enlace para la ampliación del Sistema de Alerta Temprana de Huaicos <i>Freddy Orlando Jara Poma</i>	125
Sistema de detección y control de fase remoto para un arreglo de antenas dipolo <i>César Raúl Alan Cruz Gutiérrez</i>	128
Sistema de sincronización de pulsos para vipir biestático <i>Edmundo Arom Miranda Zárate</i>	133
Sistema en FPGA para proporcionar marcas de tiempo basadas en GPS a sistemas de adquisición en el Radio Observatorio de Jicamarca <i>Víctor Eduardo Vásquez Ortiz</i>	138
Variabilidad de estrellas rápidamente oscilantes AP: HD 217522 <i>Raúl Santa Cruz</i>	142
SECCIÓN 2. Ciencias de la atmósfera e hidrósfera	142
Caracterización de la dinámica oceánica de la costa peruana bajo diferentes escenarios de esfuerzo de vientos <i>Katerine Elsy Ticse De La Torre</i>	143
Caracterización del rol del Páramo Andino en la regulación del recurso hídrico en la cuenca Ronquillo <i>Jhon Dany Orrillo Vigo</i>	148
Estudio de transporte de sedimentos en la cuenca del río Ronquillo <i>Diana Violeta Díaz Llatas</i>	152
Optimización de la técnica del riego pasivo para el control de los efectos de las heladas por modelamiento y simulación <i>Diego Renzo Ramírez Chávez</i>	157
SECCIÓN 3. Ciencias de la tierra sólida	162
Modelos de deformación cortical durante la fase intersísmica y su comparación con las velocidades GPS para la zona de subducción del Perú <i>Eliana Amelia Mantilla Calisaya</i>	168
SECCIÓN 4. Geofísica y sociedad	168
Manejo y gestión ambiental en el ecosistema manglar: estudio comparativo entre Perú – Ecuador <i>Eduard Alder Feijoo Fox</i>	169

PRÓLOGO

El Instituto Geofísico del Perú (IGP) tiene como su principal función realizar investigación científica en el campo de la geofísica, sea teórica o aplicada, pero siempre buscando su valor público como el mayor aporte que se hace a la sociedad. La información que resulta de estas investigaciones dan lugar a informes y reportes técnicos, así como a publicaciones en revistas científicas indexadas; siendo nuestro interés que la información llegue a los usuarios primarios para concluir con la cadena de generación del conocimiento. Para cumplir con esta función, el IGP siempre ha contado con un grupo importante de investigadores con diferentes especialidades en el campo de la geofísica, los mismos que además, cada año asumen la responsabilidad de formar y guiar a profesionales jóvenes en el fascinante mundo de la investigación, liderando proyectos de corta duración y cuyos resultados requieren ser publicados.

En este marco, el IGP publica el “Compendio de Estudiantes”, que desde su primer año se ha constituido como el mayor instrumento institucional que publica los aportes logrados por profesionales jóvenes durante la elaboración de sus tesis profesionales, pasantías y prácticas pre-profesionales desarrolladas en los diversos programas de investigación de la institución. De este modo, estos profesionales, con la asesoría de nuestros investigadores, logran obtener sus títulos y/o culminar sus informes de pasantías o prácticas pre-profesionales, siendo estos logros la base para formar profesionales encaminados en la investigación científica, lo cual contribuye al desarrollo de la ciencia en el país.

En este compendio, usted encontrará las investigaciones y/o desarrollo tecnológico realizados en las diversas temáticas de nuestros programas de investigación en Ciencias de la Tierra, Ciencias de la Atmósfera e Hidrósfera y Ciencias del Geoespacio y Astronomía. Esperamos que los aportes que hacen nuestros nuevos profesionales despierten su interés ya que se constituyen como claras evidencias de un futuro investigador científico. Nosotros seguimos realizando “Ciencia para protegernos, Ciencia para avanzar”.

Dr. Hernando Tavera
Presidente ejecutivo
Instituto Geofísico del Perú

EDITORES



DANNY SCIPIÓN

Doctor en Ingeniería Eléctrica y Computación de la Universidad de Oklahoma, EE.UU. Ingeniero Electrónico de la Pontificia Universidad Católica del Perú.

Actualmente es Investigador Científico del Instituto Geofísico del Perú a cargo de la Dirección Científica. Sus últimos trabajos de investigación se encuentran enfocados al desarrollo, configuración y procesamiento de datos de radares meteorológicos en el país.



JUAN CARLOS GÓMEZ

Magister en Ciencias Geológicas con mención en Geotecnia e ingeniero geólogo de la Universidad Mayor de San Marcos. Actualmente es investigador en geodinámica superficial del Instituto Geofísico del Perú, donde está a cargo de la Subdirección de Ciencias de la Tierra Sólida. Su investigación involucra el tema de la fenomenología de eventos geológicos extremos y la caracterización geodinámica de movimientos en masa en nuestro país.



YAMINA SILVA

Ing. Meteoróloga, Ph.D. en ciencias físico-matemáticas por la Universidad Estatal de Hidrometeorología de Rusia. Sus investigaciones abarcan la variabilidad climática y los eventos meteorológicos y climáticos extremos en el Perú. Ha liderado proyectos de investigación orientados a mejorar el conocimiento sobre los procesos físicos, dinámicos y microfísicos asociados a la variabilidad del clima en el Perú, principalmente en la formación de lluvias, tormentas y heladas en la cuenca del río Mantaro; para ello creó la línea de investigación e implementó el Laboratorio de Microfísica Atmosférica y Radiación-LAMAR, ubicado en el Observatorio de Huancayo del IGP.



CIENCIAS DEL GEOESPACIO Y ASTRONOMÍA

SECCIÓN 1

CARACTERIZACIÓN DE LA CAPA LÍMITE ATMOSFÉRICA EN EL VALLE DEL MANTARO EMPLEANDO RADIOSONDAS Y SENSORES REMOTOS

**Characterization of the atmospheric boundary layer in the Valle del Mantaro using radiosondas and remote sensors*

Josep Josué Prado Fasanando

Universidad Nacional José Faustino Sánchez Carrión
Facultad de Ingeniería Agraria, Industrias
Alimentarias y Ambiental
josep.prado@jro.igp.gob.pe

Asesor: Dr. Danny Scipión

Radio Observatorio de Jicamarca - IGP

RESUMEN

Se desea caracterizar la Capa Límite Atmosférica (CLA) sobre una zona con una topografía compleja como lo es el Valle del Mantaro, (Perú). Para ello se desarrolla un algoritmo que estima la altura de la CLA con datos de un radar perfilador de Capa Límite y Tropósfera (BLTR por sus siglas en inglés). El algoritmo usa datos como nivel de turbulencia y la relación señal / ruido. Estas estimaciones son comparadas y validadas con los datos obtenidos en la campaña de mediciones realizada en el Observatorio de Huancayo del IGP en febrero del 2019, donde se empleó un radar perfilador de vientos, un ceilómetro y radiosondas. Posteriormente se realiza un análisis del ciclo diurno, nocturno y climatología de la CLA con data de diciembre del 2015 hasta abril del 2019, obteniéndose que la CLA en el Valle del Mantaro presenta un régimen altamente influenciado por las gradientes de temperatura entre la superficie y la atmósfera, los procesos de turbulencia cerca de la superficie y la precipitación, lo que se podría atribuir al corto registro analizado.

Palabras clave: Capa Límite Atmosférica, Radar Perfilador de vientos, Radiosondas, Turbulencia, SNR.

ABSTRACT

Characterization of the Atmospheric Boundary Layer (ABL) over an area with a complex topography such as the Mantaro Valley, (Peru) is important to fully understand its physical parameters. For this, an estimation algorithm is developed to obtain the height of the ABL with data from a Boundary Layer and Troposphere Radar (BLTR). The algorithm uses data such as turbulence level and the signal-to-noise ratio. These estimates are compared and validated with the data obtained during the February 2019 campaign at the Huancayo Observatory of the Geophysical Institute of Perú, where a wind profiler radar, a ceilometer, and radiosondes were used. Subsequently, an analysis of the CLA's day, night and climatology cycle is carried out from December 2015 to April 2019. Obtaining that the CLA in the Mantaro Valley presents a regime highly influenced by the temperature gradients between the surface and the atmosphere, turbulence processes near the surface and precipitation, which could be attributed to the short record analyzed.

Keywords: Atmospheric Boundary Layer, Wind Profiler Radar, Radiosonde, Turbulence, SNR.

1. INTRODUCCIÓN

La investigación se centra en estudiar las características de la Capa Límite Atmosférica (CLA) en el Valle del Mantaro, debido a que la CLA tiene un rol muy importante en el estado de la calidad de aire, ya que los procesos que dentro de ella se dan establecen las condiciones y el medio bajo el cual los contaminantes emitidos se dispersan e interactúan con los demás constituyentes de la atmósfera y su medio circundante (Emeis, Schäfer, Münkel, Friedl, & Suppan, 2012). Además, el Valle del Mantaro es un territorio estratégico para realizar estudios ambientales debido a la ocurrencia de eventos meteorológicos extremos, tales como heladas, lluvias, granizadas, etc. Eventos que causan impactos negativos anualmente, generando gran impacto social y económico (Velásquez, 2005). Por ello el estudio de la CLA es de gran importancia para la comunidad científica; una de sus características más importantes es su altura, así como su evolución temporal. En los últimos años se han desarrollado técnicas para estimar la altura de la CLA, técnicas complejas que pueden mejorar la estimación y que incluyen el uso de múltiples instrumentos.

En este sentido, el Instituto Geofísico del Perú realizó la campaña de mediciones Mantaro, 2019 durante los meses de febrero y marzo en el Observatorio de Huancayo, empleando un radar perfilador de vientos y capa límite (BLTR por sus siglas en inglés) un ceilómetro y lanzamientos de radiosondas, con la información proveniente de estos instrumentos se puede estudiar las características de la CLA.

En el presente trabajo se desarrolla un algoritmo para estimar la altura de CLA, así como su evolución temporal con los datos provenientes del radar BLTR, posteriormente se comparará y validará las estimaciones realizadas por el radar con los otros instrumentos utilizados en la campaña de mediciones. Finalmente se hará un análisis del ciclo diurno, nocturno y la climatología de la CLA para toda la data existente del radar BLTR desde su ubicación y funcionamiento en el Observatorio de Huancayo.

2. DESARROLLO

2.1. Radar perfilador de vientos BLTR

EL radar perfilador de vientos BLTR (Boundary Layer and Troposphere Radar) es un radar sensible a vientos y turbulencias atmosféricas. La operación consiste en la banda Very High Frequency (VHF) que corresponde a una frecuencia de 50 MHz, a esta frecuencia el BLTR cómodamente realiza mediciones de las velocidades del viento horizontal y vertical cada 2 o 3 minutos, durante las 24 horas, es decir, proporciona información de los tres componentes del viento: zonal, meridional y vertical, este radar forma parte del Laboratorio de Microfísica Atmosférica y Radiación (LAMAR) ubicado en el Observatorio de Huancayo. La figura 1 presenta al radar BLTR.



Figura 1. Radar BLTR en el Observatorio de Huancayo

2.2. Ceilómetro

Es un instrumento que mide la altura de la base de las nubes y la visibilidad vertical empleando pulsos láser mediante la tecnología LIDAR (Light Detection and Ranging) (Markou, Bartzokas, & Kambezidis, 2007), es una herramienta ampliamente difundida para el estudio de la CLA y la calidad de aire (Emeis et. al., 2012). La figura 2 muestra una ilustración del funcionamiento del ceilómetro CHM8K empleado en esta investigación.

2.3. Radiosondas

Un radiosondeo consiste en un enorme globo inflado con gas Helio al cual se le coloca en la parte inferior una caja que contiene diversos instrumentos que a medida que el globo se va elevando, va registrando las temperaturas, humedad, viento, presión. Estas medidas son transmitidas vía radio a la estación receptora en tierra. La figura 3, muestra un perfil de Humedad Relativa obtenida del sondeo del 13 de marzo del 2019.

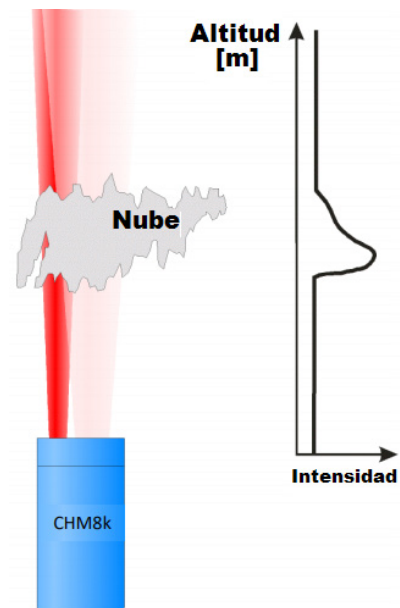


Figura 2. Ceilómetro Lufft CHM en funcionamiento

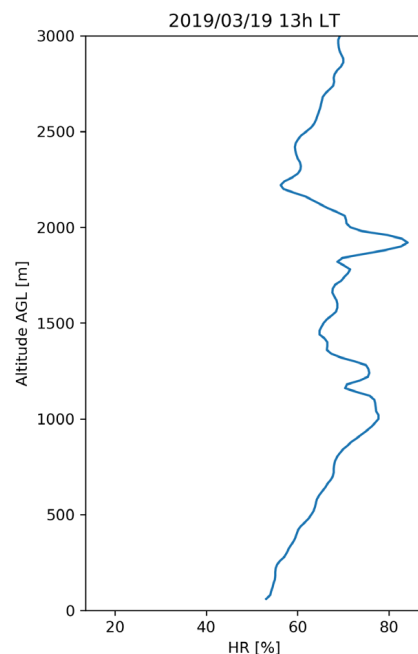


Figura 3. Perfil de Humedad relativa del radiosondeo del 13 de marzo del 2019 a las 13:00 Hora Local.

2.4. Procesamiento de datos

La programación para el procesamiento de los datos se realizó en lenguaje Python y consta de siete rutinas, las cuales se describen a continuación:

Ceilometer.py. - Es un algoritmo dedicado a la lectura de los datos del ceilómetro (archivos en formato "NetCDF"). Con esta rutina podemos escoger la variable proveniente del ceilómetro que queremos analizar, lluvia, altura de la base de las nubes, estimar la altura de la Capa Límite.

Bltr_spectra.py. - Es una rutina que permite leer y procesar los datos del radar BLTR, esta rutina nos permite escoger parámetros de las mediciones obtenidas por el radar BLTR, guardarlas y generar nuevos archivos únicamente con la o las variables seleccionadas, esta función genera archivos en formato "HDF5".

Readhdf5.py. - Es un algoritmo que permite leer los archivos generados con las variables del radar, también permite realizar gráficas con dichas variables.

Get_sonding.py. - Es una rutina dedicada a la lectura y procesamiento de los datos provenientes de los sondeos.

Get_Abl.py. - Es el algoritmo desarrollado para determinar la altura de la CLA con los datos provenientes del radar BLTR.

Validation.py. - Esta rutina realiza la comparación de las estimaciones realizadas por el radar con las estimaciones del ceilómetro y las radiosondas.

Climatology.m. - Esta rutina fue desarrollada en MATLAB, se usó la función dscatter desarrollada por Henson, (2016) la cual permite desarrollar gráficos de dispersión coloreado por la intensidad de puntos. Estos gráficos se pueden ver en las figuras 7, 8 y 9.

2.5. Metodología

Para la estimación de la altura de la CLA con las radiosondas se usaron técnicas como el Método del gradiente el cual busca detectar puntos de inflexión dominantes en los perfiles de radiosondeos analizados, en este método se analiza la temperatura potencial (θ), humedad relativa y relación de mezcla.

Para la estimación de la altura de la CLA con el ceilómetro se usan técnicas como el Método del gradiente mínimo (Hayden et al., 1997) establece que la altura de la CLA se evidencia como un mínimo en la primera derivada del perfil de intensidad de Backscatter $\delta\beta/\delta z$. Y el Método de varianza máxima, este método asume que el cambio en la concentración de aerosoles de la CLA a la atmósfera libre se traduce como un pico máximo de varianza en el perfil vertical. La construcción del perfil de varianza se realiza calculando la varianza de un grupo de datos a partir de la expresión

$$\sigma^2 = \frac{\sum_{i=1}^N n(\beta(Z_i) - \beta(Z))^2}{N} \quad 1$$

Donde $\beta(Z_i)$ es la intensidad de backscatter a la altura z y N es el número de datos tomados en cada intervalo.

Para la estimación de la altura de la CLA con el radar BLTR se tomó como referencia el trabajo presentado por Angevine, White, & Avery, (1994) el cual indica que el índice de refracción derivado del radar representado por C_n^2 alcanza su punto máximo sobre la CLA, por lo tanto un pico de la relación señal / ruido (SNR) con la corrección de rango indica el límite superior de la CLA. Basándose en esa información es que se desarrolla un algoritmo para estimar la altura de la CLA con el radar BLTR, primero el pico de SNR de cada muestra es encontrado y luego la mediana de las alturas a las cuales ocurren los picos sobre algún periodo es calculada.

2.6. Análisis estadístico

Para evaluar si la estimación de la altura de la CLA con el radar BLTR se está desarrollando correctamente, los datos serán comparados con las estimaciones obtenidas del ceilómetro y de las radiosondas. Se usaron tres indicadores estadísticos: el coeficiente de Pearson, el sesgo y el sesgo porcentual.

Coefficiente de Pearson. -Es una medida de la asociación entre dos variables. Se calcula como la razón que hay entre la covarianza de dos variables y el producto de sus desviaciones estándares, expresado como:

$$R = \frac{Cov(x,y)}{[Var(x)Var(y)]^{\frac{1}{2}}} \quad 2$$

Sesgo. - Es el indicativo de la posición de variables respecto a un valor de referencia, y si una de las variables es tomada de referencia el sesgo indica subestimación o sobreestimación con relación a la otra variable y se define como:

$$\beta = \frac{1}{n} \sum_{k=1}^n (x_k - y_k) \quad 3$$

Donde n es el número de pares de variables (mientras más cercano a 0 sea, mejor).

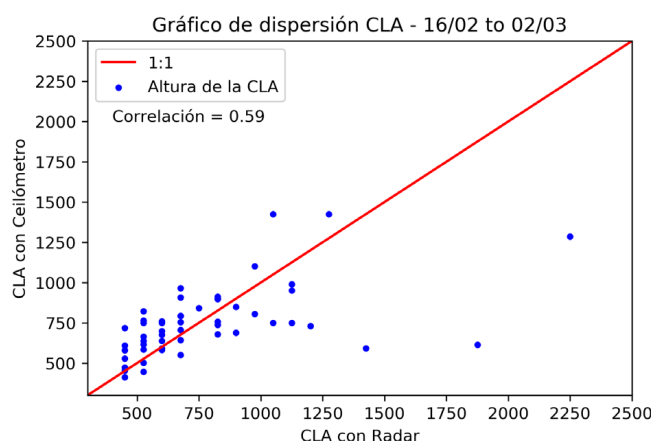


Figura 4. Gráfico de dispersión de las alturas estimadas por el radar BLTR en el eje "x" y alturas estimadas por el ceilómetro en el eje "y"

Sesgo absoluto porcentual. - La relación entre la diferencia absoluta y la variable de referencia nos indica el error sistemático promedio, el cual es calculado como:

$$|\beta_{porciento}| = \frac{\sum_{k=1}^n |(x_k - y_k)|}{\sum_{k=1}^n x_k} \quad 4$$

3. RESULTADOS

3.1. Comparación de alturas de CLA estimadas

La estimación de la altura de la CLA realizada con el programa descrito en la sección 2.4 es comparada con las estimaciones del ceilómetro y las radiosondas obteniendo un coeficiente de correlación de 0.59 en la validación con el ceilómetro (figura 4) y 0.71 con las estimaciones realizadas por las radiosondas. El sesgo es de 152.19 lo que indica que en promedio el radar

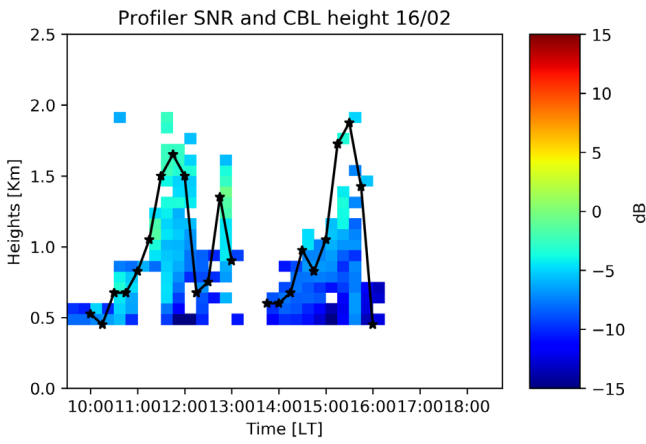


Figura 5. Altura de la CLA estimada por el radar BLTR en líneas negras, sobre el perfil de SNR del día 16 de febrero del 2019

subestima al ceilómetro en 152.19 metros y 484.15 en el caso de las radiosondas. Las diferencias entre la estimación realizada por el radar, el ceilómetro y las radiosondas puede deberse a diversos factores, entre los cuales se encuentra principalmente la estabilidad de la atmósfera; tema que debe ser profundizado en otros estudios. La figura 6 presenta la comparación entre las estimaciones realizadas por el radar y las realizadas por las radiosondas.

3.2. Ciclo diurno, nocturno y climatología

Una vez validado el algoritmo que se desarrolló se procedió con un análisis del ciclo diurno, nocturno y climatología de la CLA. Observándose que el desarrollo de la CLA coincide con el ciclo diurno del sol, esto podría indicar que la temperatura es la variable más influyente en el desarrollo de la CLA. Adicionalmente se puede observar que durante los meses enero – abril, la altura de la CLA es más baja que en los meses siguientes, lo cual coincide con la temporada de lluvias.

Es preciso señalar que pese a estar operativo desde el 2015 en el Perú, esta es la primera investigación que busca determinar las características de la CLA con el radar BLTR.

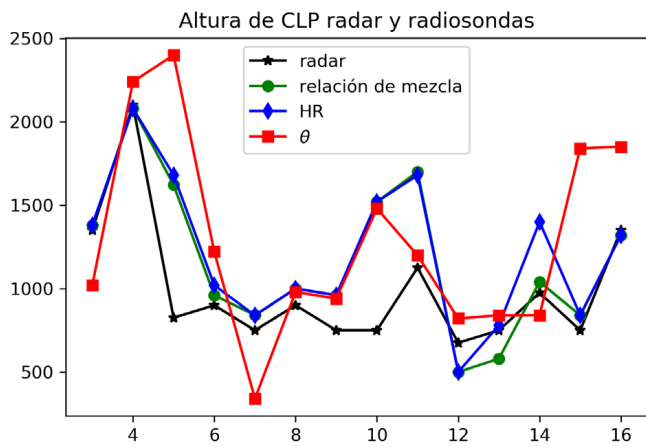


Figura 6. Comparación de alturas estimadas por el radar BLTR y las variables analizadas de las radiosondas.

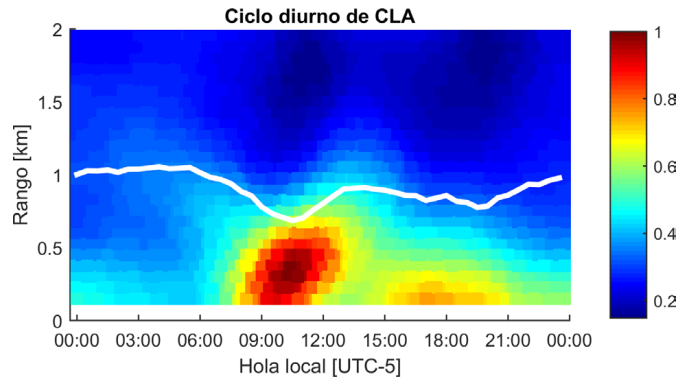


Figura 7. Ciclo diurno de la CLA en un gráfico de dispersión coloreado por intensidad, donde más cerca de 1 significa mayor cantidad de datos.

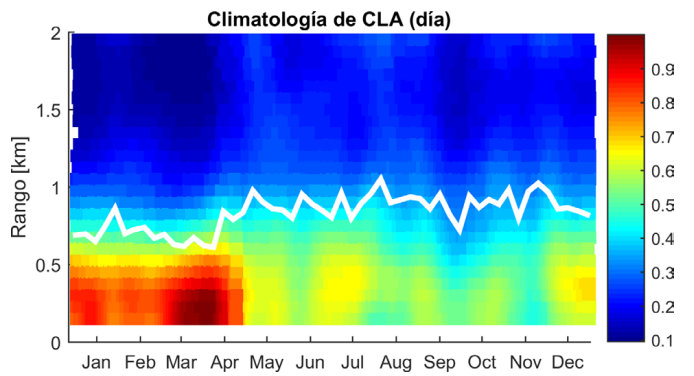


Figura 8. Climatología diurna de la CLA (07 h hasta 19h) en un gráfico de dispersión coloreado por intensidad donde más cerca de 1 significa mayor cantidad de datos.

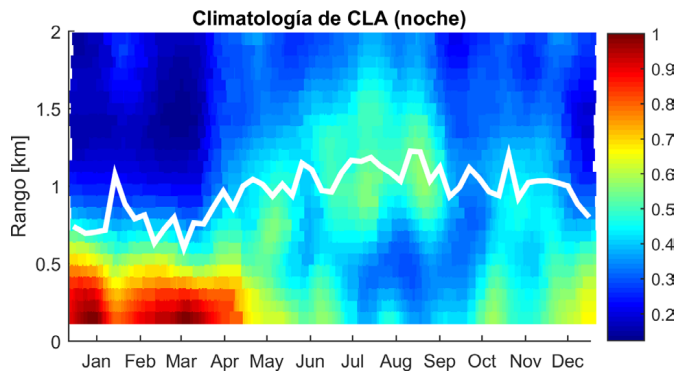


Figura 9. Climatología nocturna de la CLA (de 19h hasta las 07 horas) en un gráfico de dispersión coloreado por intensidad donde más cerca de 1 significa mayor cantidad de datos.

4. DISCUSIÓN

Para el desarrollo del algoritmo de estimación de altura de la CLA con el radar BLTR se usó la misma metodología presentada por autores como Angevine (et. al., 1994); la principal diferencia radica en que la campaña realizada en el Observatorio de Huancayo estuvo influenciada por la presencia de precipitaciones. Aun considerando que el radar BLTR tiene dos modos de operación, en este estudio se determinó estudiar principalmente la evolución de la CLA diurna, debido a que durante las horas de la noche la CLA desciende y se encuentra en alturas inferiores respecto a las que el radar tiene capacidad de medir.

5. CONCLUSIONES

El programa desarrollado para la estimación de la altura de la CLA realiza correctamente el procesamiento de la relación señal / ruido obtenida del radar BLTR y de esta forma se logra obtener la altura de la CLA con la metodología explicada en la sección 2.5. El análisis de las alturas estimadas por el radar versus los datos observados por el ceilómetro indica que existe un sesgo de 152.19 metros. Este es un error sistemático que podría tener múltiples causas, tanto instrumentales, como de procesamiento.

6. RECOMENDACIONES

Se ha demostrado que es posible usar datos del radar BLTR para estimar la altura de la CLA y su variabilidad temporal, como su climatología, sin embargo, la alta variabilidad de la CLA en el Valle del Mantaro hace que la caracterización de la CLA requiera de otros instrumentos más especializados para este tipo de estudios, como el ceilómetro usado en la campaña de mediciones.

Al encontrar que la altura en algunos meses es mayor que en otros y que en los meses de junio a setiembre se observa mayor presencia de capas remanentes podría significar que hay características peculiares de la CLA en diferentes temporadas, por lo que es necesario tener los instrumentos de referencia como el ceilómetro, por un periodo más extenso de tiempo del que se contó en esta investigación, de igual forma para mejores resultados, se recomienda mayor cantidad de radiosondeos, (véase Bianco, Wilczak & White, 2008) y revisar que las campañas realizadas para validar las estimaciones fueron más extensas e incluyeron más instrumentos.

7. AGRADECIMIENTOS

Al Instituto Geofísico del Perú por brindarme las facilidades para desarrollar mi tema de tesis.

A los doctores Danny Scipión y Yamina Silva por la orientación, amabilidad y apoyo en el desarrollo de la presente investigación.

A los doctores Renee Estevan y José Flores por su apoyo desde el Observatorio de Huancayo y finalmente al personal del Radio Observatorio de Jicamarca – IGP.

8. BIBLIOGRAFÍA

Angevine, W. M., White, A. B., & Avery, S. K. (1994). Boundary-layer depth and entrainment zone characterization with a boundary-layer profiler. <https://doi.org/https://doi.org/10.1007/BF00706797>

Bianco, L., Wilczak, J. M., & White, A. B. (2008). Convective Boundary Layer Depth Estimation from Wind Profilers : Statistical Comparison between an Automated Algorithm and Expert Estimations. International Symposium on Tropospheric Profiling, 1397–1413. <https://doi.org/10.1175/2008JTECHA981.1>

Emeis, S., Schäfer, K., Münkel, C., Friedl, R., & Suppan, P. (2012). Evaluation of the Interpretation of Ceilometer Data with RASS and Radiosonde Data. *Boundary-Layer Meteorology*. <https://doi.org/10.1007/s10546-011-9604-6>

Hayden, K. L., Anlauf, K. G., Hoff, R. M., Strapp, J. W., Bottenheim, J. W., Wiebe, H. A., ... McKendry, I. G. (1997). The vertical chemical and meteorological structure of the boundary layer in the Lower Fraser Valley during Pacific '93. *Atmospheric Environment*. [https://doi.org/10.1016/S1352-2310\(96\)00300-7](https://doi.org/10.1016/S1352-2310(96)00300-7)

Markou, M. T., Bartzokas, A., & Kambezidis, H. D. (2007). A new statistical methodology for classification of sky luminance distributions based on scan data. *Atmospheric Research*. <https://doi.org/10.1016/j.atmosres.2007.06.001>

Velásquez, V. H. (2005). Diagnóstico de la Cuenca del Mantaro bajo la visión del cambio climático: Volumen II. In *Repositorio institucional - IGP*. Retrieved from <http://repositorio.igp.gob.pe/handle/IGP/715>

CONVERSIÓN DE PROGRAMAS EN MATLAB DE V.PAZNUKOV A PYTHON PARA ESCALADO AUTOMÁTICO DE IONOGRAMAS

**Conversion of V.Paznukov programs from MatLab to Python for automatic ionogram scaling*

Luis Daniel Cárdenas Andrade

Universidad Nacional de Ingeniería
Facultad de Ciencias
Cardenasandrade273@gmail.com

**Asesores: César de la Jara
y Marco Milla**

Radio Observatorio de Jicamarca - IGP

RESUMEN

Se expone las etapas, conceptos e ideas para la conversión de un programa en MatLab de V.Pasnukov a un entorno Python 2.7. Iniciaremos con la etapa de reconocimiento de partes del programa en MatLab, esta etapa incluye estudiar el funcionamiento de cada código en el programa, entenderlo e interpretarlo en pseudocódigo. La etapa siguiente es ya la implementación del pseudocódigo en el entorno Python, en esta etapa hemos tratado de ser lo más parecido a MatLab, esto para los conocedores del código en MatLab, pero también hemos implementado algunos códigos adicionales complementando el código de MatLab. Finalizamos con la implementación de la GUI (interfaz gráfica de usuario) y el testeo del programa.

Palabras clave: conversión, reconocimiento e implementación.

ABSTRACT

The stages, concepts, and ideas for the conversion of a V.Pasnukov program from MatLab to Python 2.7 are presented. We will begin with the recognition of the structure of the program in MatLab, this includes studying the operation of each code in the program, understanding it, and interpreting it in pseudocode. Next is the implementation of the pseudocode in Python, in this stage we have tried to be as close to MatLab, this is for the connoisseurs of the code in MatLab, but we have also implemented some additional codes to complement the code from MatLab. We finish with the implementation of the GUI (graphical user interface) and the testing of the program.

Keywords: conversion, recognition and implementation.

1. INTRODUCCIÓN

El principal objetivo es implementar un programa en Python para el escalado automático de ionogramas usando el software desarrollado por V. Paznikov para la detección de trazas y el software de escalado POLAN.

Una ionosonda es un sistema de radar en la banda HF que permite realizar un sondeo de la ionósfera. Cuando se transmite una señal de radio HF de forma vertical, esta es reflejada por la ionósfera terrestre. Esta reflexión se produce a la altura en la cual la frecuencia de la señal es igual a la frecuencia del plasma (parámetro que depende directamente de la densidad de electrones del medio). Para una frecuencia diferente, la reflexión se produce a una altura diferente.

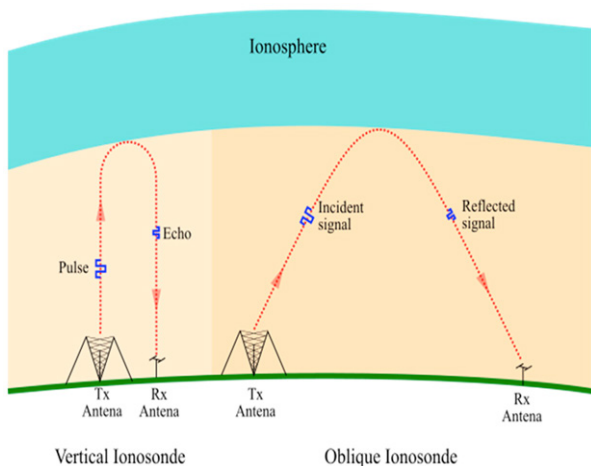


Figura 1: Funcionamiento de ionosonda.

Las ionosondas al realizar un barrido en múltiples frecuencias permiten estimar el perfil de densidad de la parte baja de la ionósfera por debajo del pico de densidad de la región F2. Para ello diferentes técnicas de procesamiento de señales para la determinación de las alturas de reflexión son empleadas. Así también, técnicas de inversión de problemas son utilizadas para poder estimar el perfil de densidad en función de las trazas detectadas en los ionogramas.

2. DESARROLLO

2.1. Características de los programas en MatLab

El programa en MatLab contiene 6 módulos los cuales son:

- findnearest.m
- my_key_pressed.m
- pro_ken_ngi_v1b.m
- randomclose3.m
- runpolan1.m
- sketch.

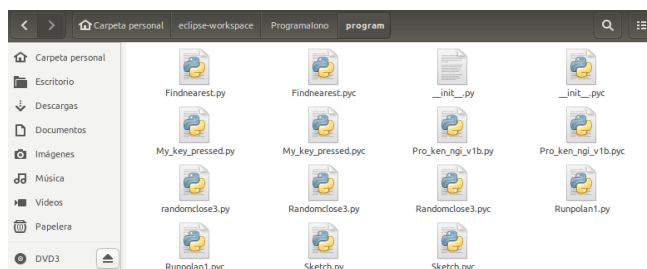


Figura 2: Directorio donde se muestra el paquete de funciones del programa en MatLab, ahora en lenguaje Python.

2.2. Desarrollo del proyecto Características

2.2.1. Contenido de los módulos desarrollados en Python 2.7

Similar al programa perteneciente al MatLab, desarrollamos una estructura parecida en el entorno Python donde presentamos las funciones principales.

- Findnearest.py
- My_key_pressed.py
- Pro_ken_ngi_v1b.py
- Randomclose3.py
- Runpolan1.py
- Sketch.py

2.2.2. Funciones principales del programa en Python 2.7

- Findnearest.py

En este módulo desarrollamos una función llamada finnearest() que busca los valores de frecuencia más cercanos a un vector de valores escalados previamente entregados.

Su input son un array de frecuencias y un array de valores escalados de frecuencia. Su output son los valores de frecuencia más cercanos a la referencia (array de valores escalados de frecuencia).

- My_key_pressed.py

En este módulo desarrollamos una función llamada GUI (root, width, height) la cual nos desarrolla una interfaz gráfica de usuario creada con la librería llamada Tkinter de Python, en este módulo agregamos en la ventana gráfica tkinter llamada root, botones los cuales son:

POLAN: ejecuta la función runpolan1(), perteneciente al módulo Runpolan1.

NEXT: ejecuta la función runpolan1(), perteneciente al módulo Runpolan1 y hace un cierre de ploteo, plt.close(), para mostrar el siguiente ionograma.

EXIT: ejecuta la función randomclose3() perteneciente al módulo Ramdonclose3. Y hace un cierre de ventana tkinter (la interfaz gráfica de usuario) con el evento [root].destroy()

- Pro_ken_ngi_v1b.py

En este módulo, de igual manera que en MatLab, se desarrolla el programa principal, su código realiza las operaciones necesarias para el acceso de datos .ngi, los cuales están en un formato HDF5, pertenecientes al sistema de almacenamiento de datos científicos netCDF, dicho acceso tiene una convención en Python, la cual explicaremos mas adelante, los cálculos matemáticos necesarios para el tratamiento de datos de ionogramas, limpieza de ruido, escalado de ejes(frecuencia y alturas), cálculos matemáticos para el auto-escalado de los datos del ionograma y en la parte final llamamos indexamos el módulo Sketch, el cual contiene a la función sketch.py donde al interfaz gráfica de usuario.

- Randomclose3.py

En este módulo desarrollamos la función llamada randomplose3 la cual se encarga del almacenamiento de datos en un archivo .dat el cual contendrá datos de frecuencia y alturas pertenecientes al output del ejecutable Polan.exe, de la misma forma que en

MatLab recalcamos que el formato de dicho archivo puede ser establecido de acuerdo a como lo decida el programador, ya que, por ahora, solo son unos datos visuales (output).

- Runpolan1.py

En este módulo se ejecuta una función llamada runpolan1() lo que hace es almacenar los datos de frecuencia y alturas, obtenidos del escalado automático y manual del ionograma, en un archivo .tmp, datos brindados por la función sketch() perteneciente a la clase Sketch.

El formato de la escritura de dichos archivos es establecido por el ejecutable Polan.exe, ya que va ser el input del mencionado ejecutable.

Luego de almacenar los datos llamará al ejecutable Polan.exe y le dará por input al archivo .tmp creado y recibirá el output del ejecutable, el cual serán frecuencias y alturas reales, después haremos una gráfica de dichos datos en la ventana tkinter.

- Sketch.py

En este módulo se encuentra la función sketch() la cual realiza el reconocimiento del evento del mouse, presionar el botón derecho (agregar valor), izquierdo (eliminar valor), movimiento del mouse (eliminar o agregar valor de acuerdo al botón presionado).

Dichos eventos pertenecen a funciones llamadas down(), up(), motion(), las cuales representan presionar un botón, soltar un botón y mover el mouse respectivamente, y generan datos los cuales son decodificados como valores de posición en la gráfica del ionograma. Esto dará ciertos valores que modificarán los datos de frecuencia y altura que son ploteados sobre el ionograma (freehands drawing) representado por una curva roja.

2.2.3. Librerías empleadas

Durante el desarrollo del proyecto, hemos empleado las siguientes librerías.

- Numpy

Para el uso matricial con equivalencias en MatLab.

- Matplotlib

Para el ploteo de graficas.

- Scipy

Usado para interpolación de un ajuste polinomial.

- netCDF4

Lectura archivos con formato HDF5.

netCDF es un formato de archivo destinado a almacenar datos científicos multidimensionales.

Esta librería contiene las herramientas necesarias para leer los datos de los archivos .ngi (ionogramas) que tenemos.

netcdf4-python es un interfaz de Python para la biblioteca netCDF. La versión 4 de netCDF, la que usamos, tiene muchas características que no se encuentran en las versiones anteriores y se implementa sobre el formato HDF5.

Usamos esta librería para el acceso a los valores de frecuencia, altura, potencia del eje extraordinario, potencia del eje ordinario, ruido en el eje extraordinario y ruido en el eje ordinario. En nuestro caso, necesitamos de una palabra clave, ya que, dicha información se almacena en un diccionario de Python, dichas palabras claves se encuentran en el código de programación que registra los datos del ionograma, en este caso la ionosonda viper. El acceso a dicha información es de la siguiente manera:

En Python

```
import netCDF4
from netCDF4 import Dataset

f = Dataset("ruta del archivo .ngi")
dato = f.variables["clave"][:] # muestra los datos del archivo .ngi
```

luego de eso tendremos los datos almacenados en la lista 'dato' y podremos proceder a hacer los cálculos respectivos con ellos.

- Tkinter

La librería Tkinter nos brinda un sencillo formato para el desarrollo de una interfaz gráfica de usuario, esta librería ya viene preinstalada en Python.

En síntesis, su uso es fácil de utilizar y su desarrollo es amigable, aunque no tiene grandes efectos visuales llamativos, lo consideramos suficiente para la realización del proyecto.

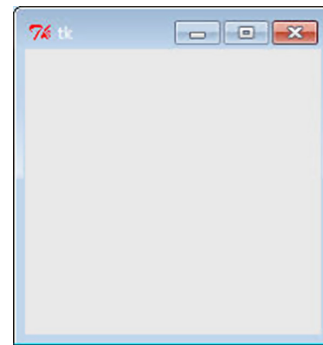


Figura 3: Ejemplo de una ventana tkinter.

- xlrd

Escritura un archivo de texto.

2.2.4. Polan.exe

Definido el input del polan en el módulo Runpolan1.py, es ingresado en el ejecutable polan.exe.

El ejecutable polan.exe recibe como parámetros frecuencias y alturas, pero no coge todas ellas, simplemente las frecuencias en un rango entre 0.50MHz y 8.900MHz, eso hace que solo se aplique el ejecutable polan.exe a las frecuencias, pertenecientes al input del polan.exe, dentro de ese rango.

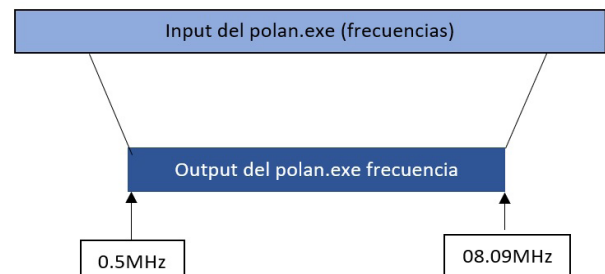


Figura 4: Esquema de comparación input and ouput del polan.exe

Por lo mencionado podemos afirmar que el polan.exe que estamos usando condiciona a los ionogramas a tener la frecuencia crítica de la capa F2 dentro del mencionado rango. Podremos percatarnos de esto si en el input brindado al polan.exe encontramos una frecuencia que sea mayor a la frecuencia

máxima brindada por el polan, esto daría a entender que la onda enviada aún se refleja a mayor frecuencia y por consiguiente tiene una mayor altura pico (peak height).

2.2.7. Implementación de un nuevo botón “AUTOMATIC POLAN”

Se implementó en el programa un nuevo botón llamado automatic polan, la función principal de este botón es calcular el escalado automático de cada ionograma, obtener un input para el polan.exe y aplicarle el ejecutable, obteniendo como resultado el output del polan.exe en un archivo .dat, para todo los ionogramas pertenecientes al archivo donde se encuentran los archivos .ngi

3. RESULTADOS

3.1. Se implementó el diseño de la GUI para el auto-escalado de ionogramas

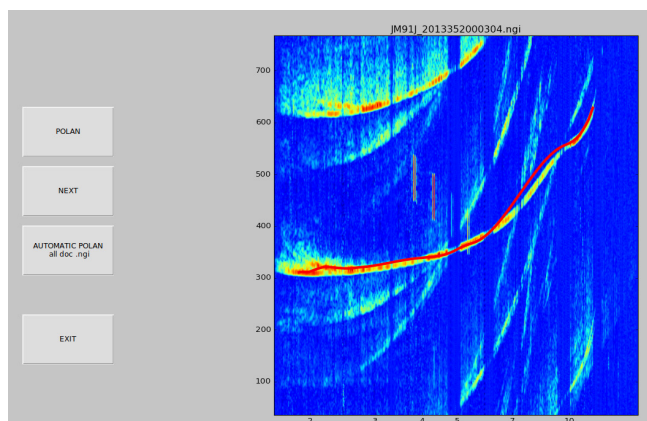


Figura 5: GUI para el ionograma Jm91J_2013352000304.ngi

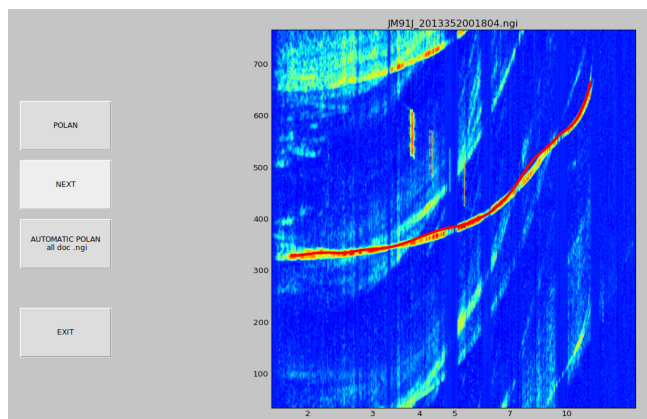


Figura 6: GUI para el ionograma Jm91J_2013352001804.ngi

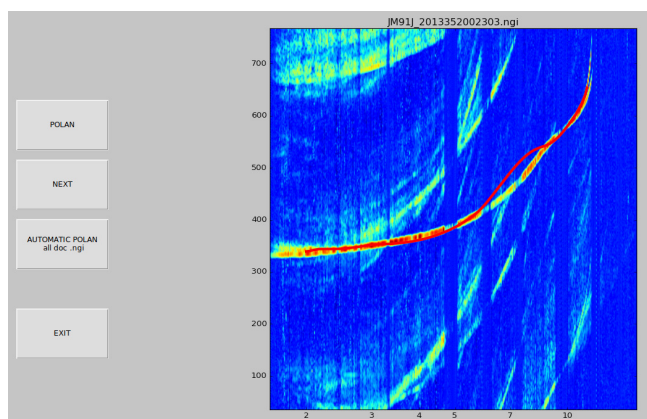


Figura 7: GUI para el ionograma Jm91J_2013352002303.ngi

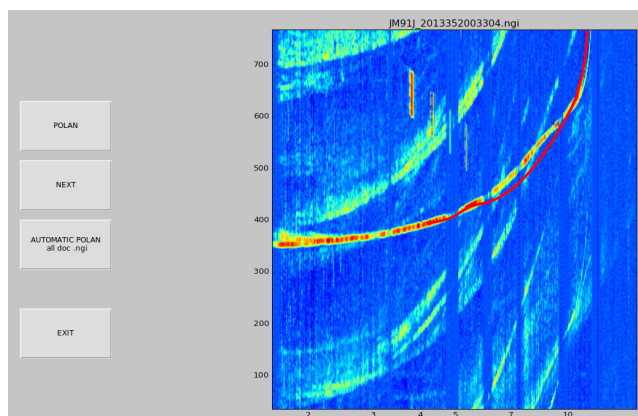


Figura 8: GUI para el ionograma Jm91J_2013352003304.ngi

3.2. Se obtuvo un archivo de texto .dat al llamar al ejecutable polan.exe

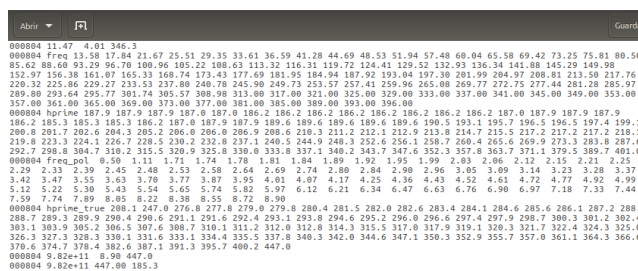


Figura 9: output del polan.exe

4. DISCUSIÓN

Para la conversión de un código de programación de un lenguaje a otro, hay que tener muy en cuenta la sintaxis equivalente entre ambos lenguajes, el método de resolución del problema de programación, y el entendimiento total del código de programación y del problema que resuelve dicho código.

En el proyecto se tomó los primeros días para instalar los programas necesarios para la realización del proyecto, MatLab y Python, se eligió una IDE para la programación en Python, Eclipse oxígeno, se fijó claramente que resolvía, escalado automático y manual de ionogramas, y cómo funcionaban las distintas funciones en el programa de MatLab.

Uno de los mayores retos presentados en el proyecto fue la sintaxis y la equivalencia de funciones entre MatLab y Python, tuvimos que modificar algunas líneas de código e inclusive crear algunas funciones que hagan lo mismo que una función predeterminada en MatLab.

Se tomó en cuenta la estructura del programa y se realizó una sintaxis parecida al del MatLab, esto por si alguien a futuro desea modificarlo, conociendo el programa en MatLab, lo pueda hacer sin mucho problema.

Se utilizó una interfaz gráfica de usuario con una librería preinstalada en Python, Tkinter, esto debido a que la GUI que queríamos obtener no mostraba gran complejidad en gráficos y porque Tkinter es muy fácil de usar.

5. CONCLUSIONES

Se logró la conversión del código, del programa de MatLab para la detección de trazas de V. Pasnikov, al lenguaje de programación Python.

Se logró la implementación de una GUI en Python para el procesamiento manual de trazas.

Se logró implementar un nuevo botón para el control automático de todos los archivos .ngi contenidos en un mismo directorio ejecutándolos con el polan.exe y recibiendo el output de cada uno de ellos.

Se logró la integración de la GUI con POLAN para la estimación automática de los perfiles de densidad.

6. RECOMENDACIONES

Se recomienda plantear una interfaz gráfica de usuario con alguna otra librería más compleja, en donde quizá podamos encontrar algunas rutas más eficientes de las que tenemos.

Se recomienda quizá, corregir algunos comandos para el uso de este programa en Python 3, para así actualizar el intérprete Python.

En el ploteo se recomienda la implementación de un ploteo más eficiente que no consuma mucho tiempo en plotear cada ionograma en cada evento de cambio manual.

Se recomienda evaluar las posibilidades de usar un nuevo ejecutable POLAN en Ubuntu, o en su defecto cambiar la extensión .exe del ejecutable del polan a una que se pueda ejecutar en UBUNTU.

7. AGRADECIMIENTOS

Deseo dirigir mi agradecimiento al Instituto Geofísico del Perú, a través del Radio Observatorio de Jicamarca, por haberme dado la oportunidad de pertenecer a su Programa de Prácticas Pre-profesionales.

A su vez, agradecer a mis asesores, el Ing. César de la Jara y el Dr. Marco Milla y también a la Ing. Karim Kuyeng y al Tesista José Suclupe por haberme guiado y haber resuelto todas mis dudas durante el desarrollo de mi proyecto.

8. BIBLIOGRAFÍA

[1] Documentación del interfaz netCDF4

http://nco.sourceforge.net/nco.html#netCDF2_002f3_002f4-and-HDF4_002f5Support

[2] Documentación de la librería Tkinter para GUI.

<https://media.readthedocs.org/pdf/guia-tkinter/latest/guia-tkinter.pdf>

[3] Documentación Python2.7

<https://docs.python.org/2/index.html>

[4] Documentación MatLab

<https://la.mathworks.com/help/matlab/?requestedDomain=true>

DESARROLLO DE UN SISTEMA DE ALARMA INTELIGENTE PARA EL MONITOREO DE EXPERIMENTOS CON EL RADAR PRINCIPAL

**Development of an intelligent alarm system for the monitoring of experiments with the main radar*

Nicol Fiorella Huamán Molina

Universidad Católica San Pablo
Facultad de Ingeniería y Computación
nicol.huaman@ucsp.edu.pe

Asesor: Juan Carlos Espinoza

Radio Observatorio de Jicamarca - IGP

RESUMEN

En el presente informe se presenta el desarrollo de un sistema de alarma para notificar de algún error producido en el radar principal del Radio Observatorio de Jicamarca. Este aviso se realizará de tres maneras diferentes ya sea sonora, lumínica y por envío de correo electrónico para informar al personal de turno del error para que actúen de inmediato. La programación se realizará utilizando el software Energía propio de la tarjeta Tiva TM4C1294 de Texas Instruments, luego se desarrollará la API REST para la configuración y organización de los comandos para que finalmente se pueda controlar mediante un terminal utilizando lenguaje Python.

Palabras clave: [Alarma] [Radar] [Energia] [Tiva] [API] [REST] [Python]

ABSTRACT

This report presents the development of an alarm system to notify of an error in the main radar of the Jicamarca Observatory Radio, scientific facility of the Geophysical Institute of Peru. This notice will be made in three different ways: sound, light, and by sending email to inform the staff of the error so they act immediately. The programming will be done using the own Energy software of the Tiva TM4C1294 card of Texas Instruments, then the REST API will be developed for the configuration and organization of the commands so that it can finally be controlled by a terminal using Python.

Keywords: [Alarm] [Radar] [Energia] [Tiva] [API] [REST] [Python]

1. INTRODUCCIÓN

El radar principal es uno de los equipos más importantes con los que cuenta el Radio Observatorio de Jicamarca (ROJ), permitiendo la observación de los fenómenos ionosféricos sobre nuestro territorio.

Muchos experimentos necesitan cierta vigilancia si es que ocurre algún inconveniente como capacidad insuficiente porque la memoria se llenó o algún otro error del software de procesamiento (Signal Chain). Estos errores deben de informarse de inmediato al personal que se encuentre supervisando en ese momento para una pronta corrección de tales errores.

Es por eso que se necesita la implementación de un sistema de alarma inteligente que pueda notificar al instante el acontecimiento de alguna falla en la operación del radar principal.

El desarrollo de dicho sistema contará con el uso de un sistema embebido el cual se encontrará conectado a la red para que así pueda notificar del error por mensajes, por correo electrónico, además de avisar sonoramente utilizando un buzzer y mostrando una señal lumínica establecida por leds. El microcontrolador utilizado es TIVA C modelo TM4C129EXL, el cual es programado por el software Energía, para luego realizar el control y la activación de alarmas mediante un terminal en el que se establece comandos desarrollados en una API REST los cuales serán ejecutados por lenguaje Python.

2. DESARROLLO

2.1. Diseño teórico del sistema de alarma

Cuando existe una falla en el radar principal, este comunica al microcontrolador del error, el cual se encuentra programado con la API REST realizada, una vez que la alarma llega al microcontrolador, este emite los tres tipos de alarma: alarma lumínica, alarma

sonora y notificación por correo electrónico para que finalmente el operador pueda controlar el sistema mediante Python.

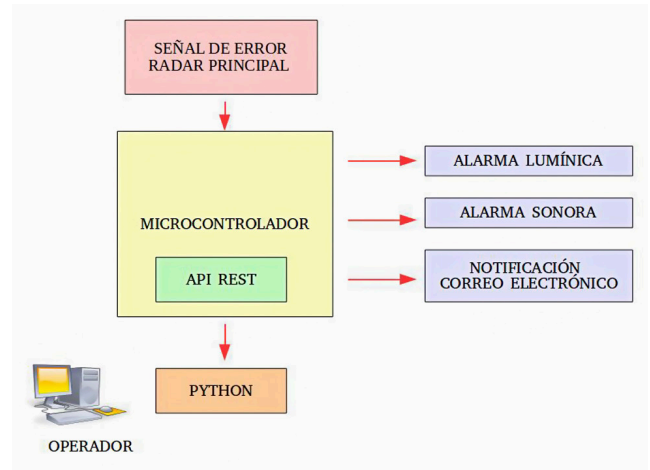


Figura 1. Diseño teórico del sistema de alarma

2.2. Implementación del sistema de alarma

La programación se realiza utilizando el software Energía, este programa trabaja con el embebido elegido. [1]

Primero se ejecuta la programación para el envío de mensajes, para este paso, el microcontrolador tiene que estar conectado a un router, el cual nos permitirá la conexión de red. Se establece un email servidor que corresponda al grupo de jro.gob.pe, la extensión es la siguiente: jro-zimbra.igp.gob.pe.

Luego se colocan los parámetros de red, es decir, ip, gateway, subnet y la mac. Una vez establecidos estos datos se colocan cada una de las salidas correspondientes como: buzzers y los leds.

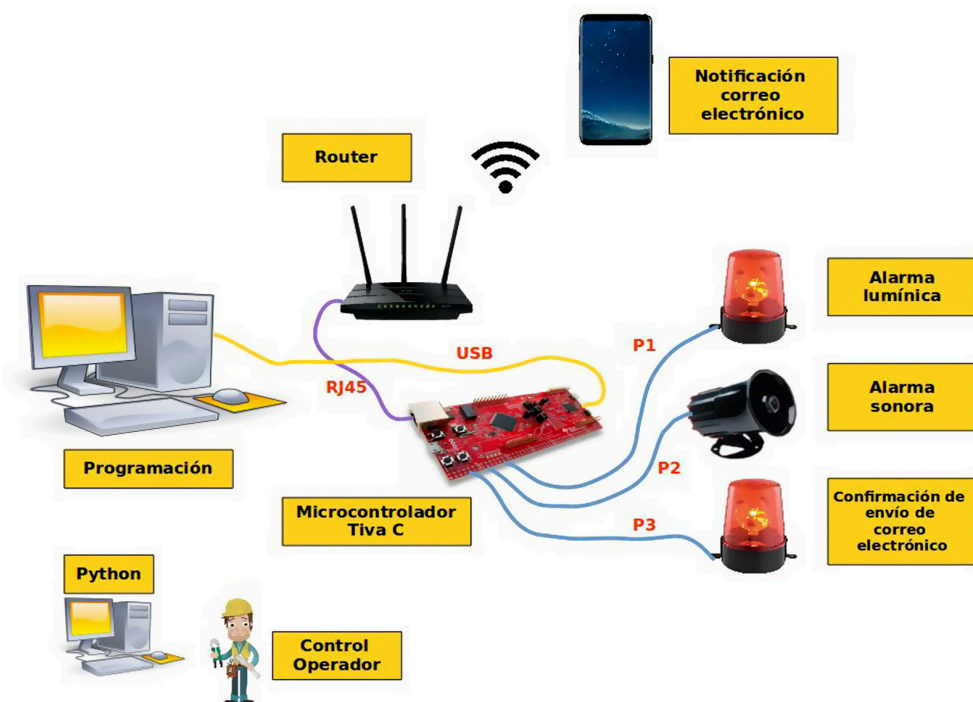


Figura 2. Diseño del sistema de alarma

Después de haber colocado los datos iniciales, se establece un comando que simulará un error al momento de colocarlo en el Monitor Serial de Energía.

Seguidamente, se hace una verificación de conexión si es que el embebido y el router pudieron conectarse. El tiempo límite de espera es de 10 segundos.

Una vez establecida la conexión, se colocan los datos de mensajería, como: remitente, destinatario, el asunto y finalmente, el mensaje.

2.3. Desarrollo de la API REST

Hasta este punto se ha realizado la programación en *Energía*, el siguiente paso es desarrollar la API REST.

Para ello se desarrolla una tabla con los métodos que uno desee según la necesidad del programador.

En nuestro caso establecimos dos tipos de formato JSON: GET y POST.

Del tipo GET se dispuso el comando "status", el cual leerá los valores actuales que han sido configurados.

Del tipo POST se dispuso los comandos: "setalarm", "changeip" y "stopalarm". Los cuales podrán ser configurados y leídos por el procesador.

Una vez establecidos los comandos que se utilizarán, se comenzará con la edición del código RC_Rest_server que fue compartido por los asesores del programa que realizaron otros proyectos similares a éste.

Request	Type	Comments
status	Get	Verifica la conexión entre el microcontrolador y la aplicación REST
changeip	Post	Configura nuevos parámetros de red para una nueva zona de instalación
setalarm	Post	Configura los parámetros del tipo de alarma (sono-ra, luz, notificación)
stopalarm	Post	Configura parámetro para apagado de alarma

Tabla 1. Comandos REST

3. RESULTADOS

3.1. Diseño teórico del sistema de alarma

3.1.1. Elección del microcontrolador

De acuerdo a las características deseadas se eligió el siguiente:

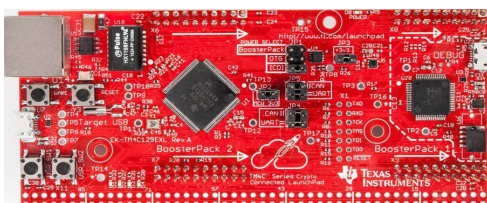


Figura 3. Microcontrolador elegido Tiva TM4C129EXL

A continuación se enumeran las características con las que se justifica la elección:

- Microcontrolador TM4C129ENCPDT
- Conectividad Ethernet con modulación de ancho de pulso (PWM) de control de movimiento 10/100 Ethernet MAC y PHY totalmente integrada
- Conector USB 2.0 Micro A / B
- Cuatro LEDs de usuario
- Dos botones de usuario
- Un interruptor de despertador de hibernación independiente
- Un interruptor de reinicio del microcontrolador independiente
- E / S traído al borde del tablero para expansión de tablero.

3.2. Implementación del sistema de alarma

En la siguiente figura se muestra las conexiones realizadas de acuerdo al sistema de alarma establecido en el capítulo anterior.

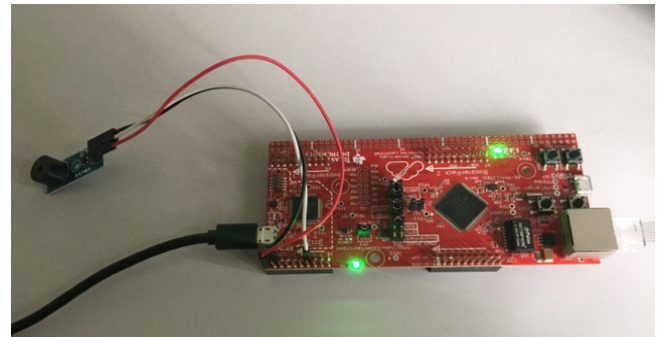


Figura 4. Implementación física del sistema de alarma

Como se presentó en la sección anterior de Desarrollo, se realizó la programación en *Energía*, los resultados que son mostrados a continuación pertenecen al *Monitor Serial* del mismo software.

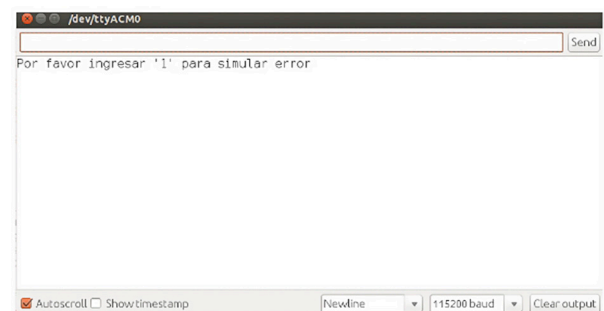


Figura 5. Simulación de error por comando

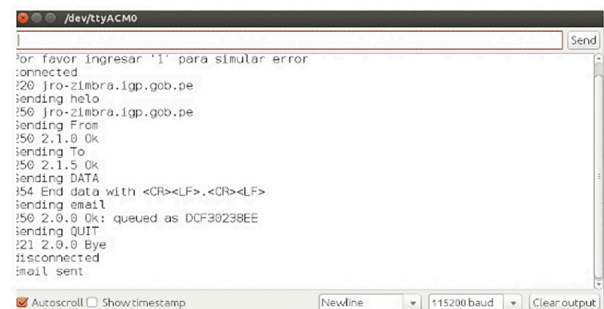


Figura 6. Conexión correcta del envío de correo electrónico

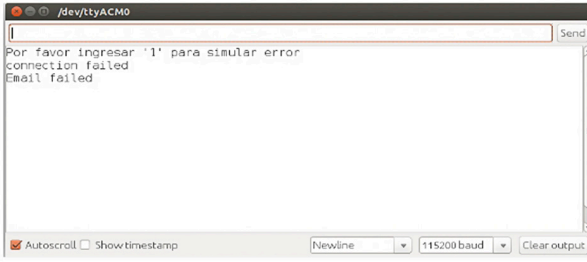


Figura 7. Conexión incorrecta del envío de correo electrónico

Después de mostrar la confirmación, en el correo electrónico del usuario se muestra de la siguiente manera:

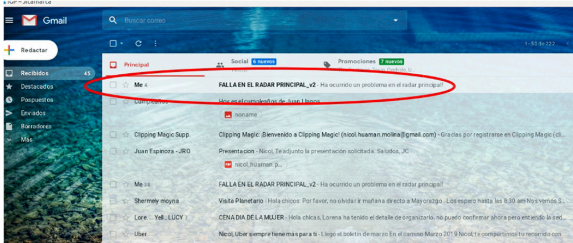


Figura 8. Envío del correo electrónico

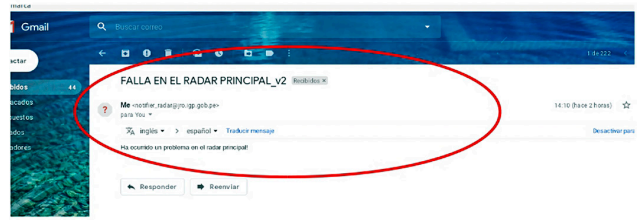


Figura 9. Correo electrónico enviado visto desde una PC

3.3. Desarrollo del API REST

En la sección anterior se determinó los comandos que se utilizarán para este proyecto, seguidamente se ejecutarán en Python a través del terminal para verificar su funcionamiento [2]. Los resultados de las pruebas se muestran a continuación:

```

nhuaman@nhuaman-To-be-filled-by-0-E-M:~$ python
Python 3.7.1 (default, Dec 14 2018, 19:28:38)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import requests
>>> r=requests.get('http://10.10.10.37/status')
>>> r.json()
{'status': 'enable'}
>>> r=requests.post('http://10.10.10.37/stopalarm', data={'command':1})
>>> r.json()
{'state': 'stop'}
>>> r=requests.post('http://10.10.10.37/stopalarm', data={"command":0})
>>> r.json()
{'state': 'sin cambios'}
>>> r=requests.post('http://10.10.10.37/setalarm', data={"option":1})
>>> r.json()
{'option': 'sound OK'}
>>> r=requests.post('http://10.10.10.37/setalarm', data={"option":2})
>>> r.json()
{'option': 'sound - email OK'}
>>> r=requests.post('http://10.10.10.37/setalarm', data={"option":3})
>>> r.json()
{'option': 'email OK'}
>>> r=requests.post('http://10.10.10.37/setalarm', data={"option":4})
>>> r.json()
{'option': 'light OK'}
>>> r=requests.post('http://10.10.10.37/changeip', data={"ip":[10,10,10,30],"dns":["10,10,10,1","gateway":[10,10,10,1],"subnet":[255,255,255,0]})
>>> r.json()
{'changeip': 'ok'}
>>>
    
```

Figura 10. Comandos REST

En la figura 10 se puede observar el control de cada uno de los métodos creados para nuestra aplicación del sistema de alarma, con cada una de las opciones establecidas de acuerdo a la necesidad del operador.

4. DISCUSIÓN

De acuerdo a las librerías utilizadas en el software *Energía* es que se pudo realizar tanto la comunicación vía Ethernet como la comunicación por comandos desde el terminal con Python. Esto nos demuestra la eficiencia que tuvo el usar la tarjeta Tiva TM4C1294 como microcontrolador.

Los métodos establecidos en la API REST en este proyecto son base para futuros dispositivos que necesiten de un sistema de alarma

ya que se pueden crear más de ellos o mejorarlos colocándoles más parámetros, haciendo más eficiente el control de ellos.

5. CONCLUSIONES

Para la realización del proyecto se tuvo que realizar un estudio de los equipos a utilizar, además del lenguaje requerido por ellos para la programación deseada.

El microcontrolador utilizado simplificó el uso de conexiones ya que en ella se encuentran leds que se utilizaron como avisos, la implementación del envío de notificación por correo electrónico y alarma de forma lumínica.

Se pudo establecer conexión con la red mediante programación en *Energía*, utilizando las librerías correctas y haciendo uso de un

servidor local autenticado otorgado por el área de soporte del Radio Observatorio de Jicamarca.

Finalmente, el control desde el usuario se desarrolló en Python mediante los comandos establecidos para cada una de las opciones de alerta.

6. RECOMENDACIONES

Se pueden establecer más comandos para tener más opciones en el control de la alarma, como por ejemplo: intervalo de tiempo de la alarma sonora, etc.

En cada método se puede colocar más opciones si es que se requiere hacer el programa con más alternativas de selección.

Para el envío de correo electrónico se puede crear una lista de correos con una dirección específica para los usuarios interesados.

Para la prueba de los comandos de control se ha utilizado Python, pero para un control más organizado se puede reutilizar la aplicación en "Postman".

El sistema se puede integrar con otros dispositivos desarrollados en el ROJ, como es el caso del transmisor ATRAD realizado por otro practicante del mismo programa, el cual puede alertar de alguna falla en este.

7. AGRADECIMIENTOS

Agradezco al Instituto Geofísico del Perú por permitirme ser parte del programa de prácticas pre-profesionales el presente año, en el que he podido aplicar mis conocimientos dictados en mi Universidad para un proyecto del Radio Observatorio de Jicamarca.

Un agradecimiento muy especial a mi asesor Juan Carlos Espinoza por su guía durante la realización del proyecto y a cada una de las personas que se encuentran laborando en el ROJ por su amable disposición de servicio y su espíritu de compañerismo al compartir sus conocimientos con nosotros.

8. BIBLIOGRAFÍA

[1] Energía's manual. Recuperado de: <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>

[2] Luna Jaramillo, Juan, "Generados de pulsos sincronizados con GPS usando API REST,"

DESARROLLO DE UNA INTERFAZ SOFTWARE BASADA EN GNU RADIO PARA CONTROLAR EL SISTEMA DE ADQUISICIÓN JARS 2.0 DEL RADIO OBSERVATORIO DE JICAMARCA

**Development of a software interface based on gnu radio to control the jars 2.0 Acquisition system of the jicamarca radio observatory*

Isaac Mario Tupac Dávila

Universidad Nacional Mayor de San Marcos
Facultad de Ingeniería Electrónica, Eléctrica y Telecomunicaciones
isaac.tupac@jro.igp.gob.pe

Asesores: Ing. John Kerim Rojas Paredes, Dr. Marco Milla Bravo

Área de investigación, desarrollo e innovación I&D+I
Radio Observatorio de Jicamarca - IGP

RESUMEN

El sistema de adquisición del Radio Observatorio de Jicamarca, conocido como JARS (Jicamarca Acquisition Radar System), ha pasado desde sus inicios por varias etapas: desde su implementación con un solo canal hasta la versión que trabaja con ocho canales (JARS 1.0). Actualmente se tiene diseñado un nuevo sistema de adquisición denominado JARS 2.0. Para el control de este sistema de adquisición se ha desarrollado un software llamado: JROAdquisicion. Este software está programado en lenguajes C y C++. Para estandarizar el software de control del JARS 2.0 utilizaremos GNU Radio. GNU Radio nos proporciona de un conjunto de herramientas para el desarrollo de software en código abierto y libre. Además, nos proporciona de clases y funciones basadas en lenguaje C++ que nos permitirán poder implementar una interfaz software entre el software JROAdquisicion y el software GNU Radio. Además de esto, obtendremos la estandarización de la data adquirida en el formato `digital_rf`.

Palabras clave: JARS 2.0, JROAdquisicion, GNU Radio.

ABSTRACT

The Jicamarca Radio Observatory radar acquisition system, known as JARS (Jicamarca Acquisition Radar System), has passed since the beginning for several stages: since its implementation with only one channel from the version that works with eight channels (JARS 1.0). Nowadays, there is a new version, JARS 2.0 currently under development. JROAdquisicion is the software used to control the acquisition. This software is programmed in C and C++ language. To make standard the JARS 2.0 control software, we will use GNU Radio. GNU Radio provides a toolkit to the develop of free and open source software. Furthermore, GNU Radio provides classes and functions based on C++ language which allow us implement a software interface between the software JROAdquisicion and the software GNU Radio. Additionally, we will obtain the standardization of the acquired data in the `digital_rf` format.

Keywords: JARS 2.0, JROAdquisicion, GNU Radio.

1. INTRODUCCIÓN

Para la adquisición de la data de la ionósfera ecuatorial el Radio Observatorio de Jicamarca tiene diseñado un sistema de adquisición propio denominado JARS (Jicamarca Acquisition Radar System). La versión actual que tiene operativa es el JARS 1.0, el cual presenta hasta ocho canales de adquisición con un ancho de banda de 1MHz cuando se utilizan todos sus canales.

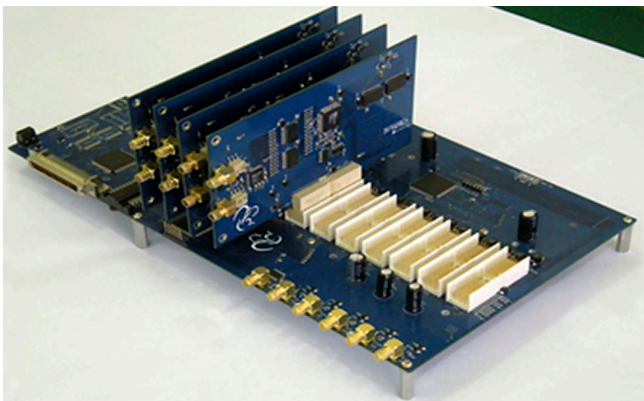


Figura 1: Sistema de Adquisición JARS 1.0

En la actualidad se tiene diseñado e implementado un nuevo sistema de adquisición llamado JARS 2.0, el cual está compuesto de cuatro tarjetas principales las cuales son: tarjeta de adquisición, tarjeta de bus, tarjeta de control y tarjeta de distribución de reloj.

Considerando que el JARS 2.0 está implementado para adquirir mayor cantidad de muestras, la velocidad de transmisión de los bits entre tarjetas debe ser mayor, por lo que se utiliza la interfaz física SATA entre la tarjeta de bus y la tarjeta de control bajo el estándar LVDS. Otra característica de este sistema de adquisición es que, si trabajamos con los 8 canales de adquisición, el ancho de banda por canal puede llegar hasta los 2 MHz y la velocidad de transmisión de los bits hacia la computadora de adquisición sería de 512 Mbps.

Se utilizará el protocolo UDP para transmitir las muestras y los comandos de configuración del JARS 2.0. También, se utilizará como interfaz física para esta transmisión la conexión Gigabit Ethernet entre la tarjeta de control y la computadora.

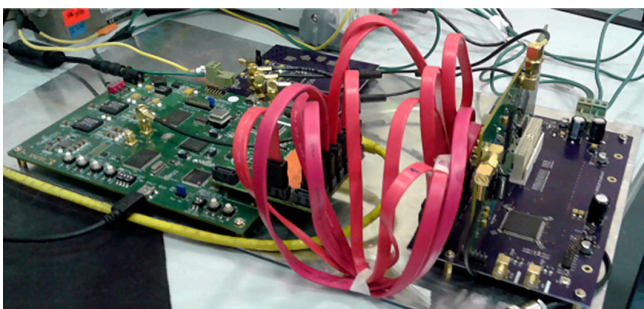


Figura 2: Sistema de Adquisición JARS 2.0

Este nuevo sistema de adquisición es configurado a través de un software denominado JROAdquisicion, el cual está programado en lenguaje C y C++.

Se tiene como objetivo que este nuevo sistema de adquisición sea parte de un sistema de radio definido por software (SDR). Así que para estandarizar el software de configuración usaremos GNU Radio, el cual nos permitirá desarrollar una interfaz con el software de adquisición JROAdquisicion, usando las API's que nos proporciona.

2. FUNDAMENTOS TEÓRICOS

2.1. Software Defined Radio

Una o varias etapas que conforman un sistema de radiocomunicación son implementadas y configuradas por software.

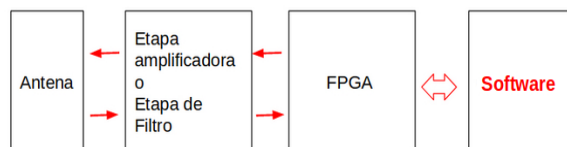


Figura 3: Sistema de Radio Definido por Software

2.2. GNU Radio

Es un conjunto de herramientas para el desarrollo de software. Nos permite implementar un sistema de radio definido por software (SDR) a través de los bloques de procesamiento de señales que nos proporciona. Estos bloques de procesamiento están guardados en los módulos por los que está formado GNU Radio.

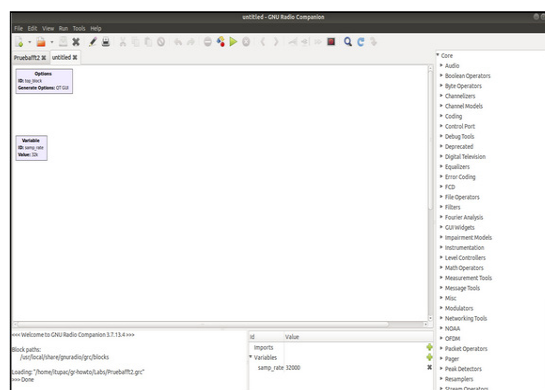


Figura 4: GNU Radio-Companion

2.3. USRP

Es un periférico universal de un sistema de radio definido por software. Podríamos decir que es el enlace entre lo real (la señal) y el software (GNU Radio).

2.2.1. Estructura interna de una USRP

- Correcciones I/Q (IQC): bloques funcionales dentro del FPGA para compensar el desbalance de las señales I/Q
- DEC/INT: bloques de decimación e interpolación
- DUC/DDC: Digital Up-Conversion Digital (DUC) y Down-Conversion (DDC). Ajusta y obtiene una frecuencia más precisa.
- UHD: El UHD es el driver el cual nos permite configurar y controlar una USRP.

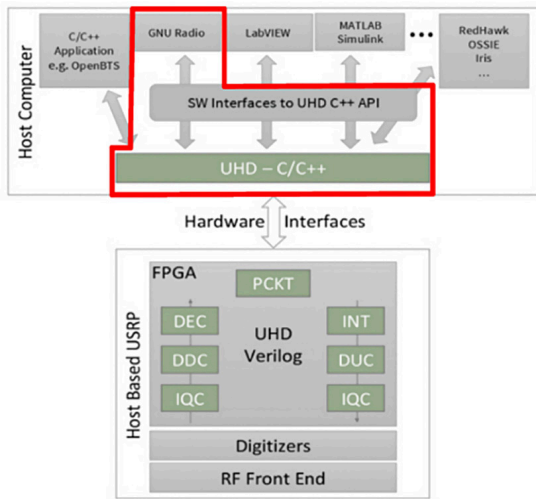


Figura 5: Estructura interna de una USRP

En la parte del software, vemos que se ha implementado una interfaz software entre el GNU Radio y el driver del USRP, llamado UHD, usando API's del UHD basadas en C++.

3. DESARROLLO DE LA TESIS

3.1. Planteamiento de la tesis

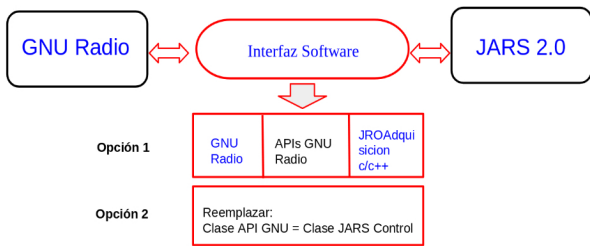


Figura 6: Estructura interna de una USROP

Se ha planteado dos opciones para el desarrollo de esta interfaz:

Opción 1: Desarrollar una interfaz entre GNU Radio y el software de adquisición JROAdquisicion basándonos en las API's de GNU Radio.

Opción 2: Utilizar los métodos y las funciones que nos provee las API's de GNU Radio para reemplazar las clases del software JROAdquisicion.

3.2. Generación de módulos Out-of-three

Se denomina módulos Out-of-three a los módulos personalizados creados en GNU Radio. Al ser personalizados no pertenecen a los módulos principales que provee GNU Radio, por eso el nombre de Out-of-three.

GNU Radio nos provee de un comando denominado: **gr_modtool**, el cual nos permite especificar y crear un nuevo módulo. Dentro de estos módulos crearemos los bloques personalizados. Éstos pueden estar basados en lenguaje C++ o Python.

Cada bloque estará definido por dos códigos principales:

- Código fuente: Este código implementa el procedimiento que deben seguir los valores de entrada para originar los valores de salida.
- Código de verificación: Este código está implementado por los valores de entrada y los valores de salida previstos. Verifica que el código fuente produzca los valores de salida previstos.

El comando **gr_modtool** también generará los bloques en la interfaz gráfica GNU Radio companion, los cuales tienen una extensión .xml

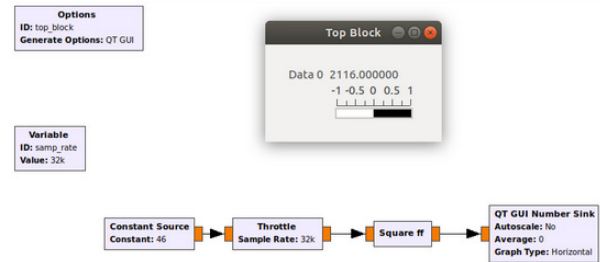


Figura 7: Generación del bloque Square ff

En la figura 9 creamos el bloque denominado: Square ff, el cual recibe un número en la entrada y origina el cuadrado del número a la salida.

3.3. JROAdquisicion

JROAdquisicion es el software de control del sistema de adquisición JARS 2.0. Está dividido jerárquicamente y cada parte cumple una función determinada.

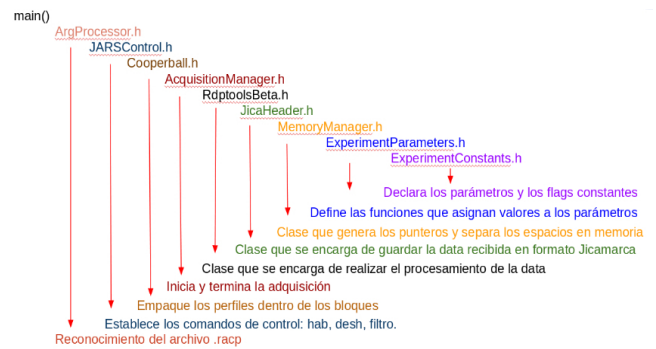


Figura 8: Estructura del software JROAdquisicion

- ArgProcessor: Reconoce el archivo .racp
- JARSControl: Establece los comandos de control: Habilitación, deshabilitación, filtros. Realiza el proceso de adquisición de bloques.
- Cooperball: Genera bloques de datos mediante funciones pseudoaleatorias para simulación.
- Acquisition Manager: Inicia y termina la adquisición.
- RdptoolsBeta: Clase que se encarga de realizar el procesamiento de la data.
- JicaHeader: Clase que se encarga de guardar la data recibida en formato Jicamarca.
- MemoryManager: Clase que genera los punteros y separa los espacios en memoria.
- ExperimentParameters: Define las funciones que asignan valores a los parámetros obtenidos del archivo .racp
- ExperimentConstants: Declara los parámetros y flags constantes.

3.4. Estructura de la data recibida

El JARS 2.0 recibirá la data adquirida a través de una estructura predefinida. El empaquetamiento y especificación de cada bloque de data se implementa por medio del software JROAdquisicion. El tipo de data obtenida puede ser: data cruda o data procesada.

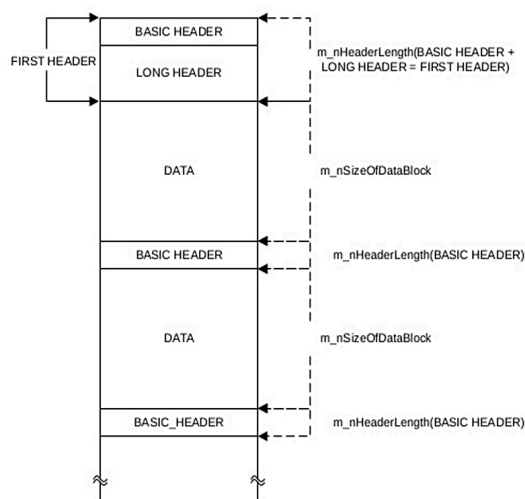


Figura 9: Estructura de la data cruda recibida.

- Basic Header: Contiene la información temporal del bloque de dato que sigue al basic header.
- Long Header: Contiene los parámetros de procesamiento y del radar. Se subdivide en tres partes: System Parameters, Radar Controller Parameters y Process Parameter.
 - System Parameters: Contiene las variables del sistema de adquisición como el número de muestras de la adquisición.
 - Radar Controller Parameters: Contiene los parámetros del controlador de radar.
 - Process Parameter: Contiene los parámetros de procesamiento aplicado a la data.
- Bloque de dato (Data): Este bloque guardará la cantidad total de perfiles y dentro de cada perfil estarán las muestras adquiridas, dependiendo del tipo de formato, pudiendo ser: data cruda o data procesada.

Cada una de estas partes será definida con sus respectivos parámetros a través del software JROAdquisicion.

Cada uno de estos bloques de data guardará una determinada cantidad de perfiles dependiendo del experimento y cada perfil guardará una determinada cantidad de muestras solicitadas.

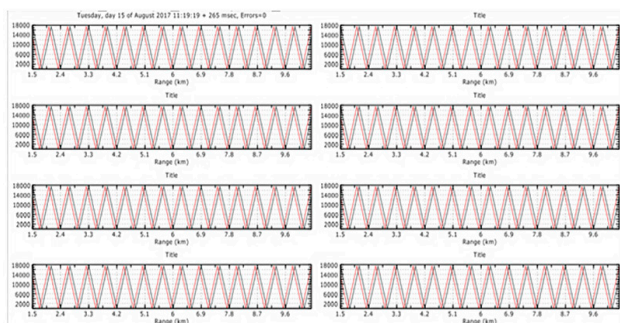


Figura 10: Ocho perfiles. Cada perfil nos da las muestras adquiridas

6. DISCUSIÓN

Con respecto a las opciones planteadas para el desarrollo de esta interfaz, se tomará la decisión después de identificar y analizar el software de adquisición JROAdquisicion y las API's que proporciona GNU Radio.

Con respecto a la generación de módulos Out-of-three, se planteará generar un bloque que reciba todos los parámetros

requeridos por un experimento y que pueda ser reconocida por la interfaz para poder comunicarse con el JROAdquisicion.

Se estudiará si GNU radio puede reconocer el archivo de texto. racp que contiene todos los parámetros que reconoce el JROAdquisicion.

7. CONCLUSIONES

Esta tesis buscará la estandarización del software de control del JARS 2.0 mediante GNU Radio.

GNU Radio nos ofrece el comando gr_modtool para la creación de nuevos bloques de una manera fácil ya que además crea todas las carpetas necesarias para su reconocimiento.

Para la elaboración de nuestros bloques con un mayor rendimiento en GNU Radio usaremos el lenguaje C++.

Estandarizaremos la data obtenida al formato digital_rf

8. RECOMENDACIONES

Definir qué opción se usará para el desarrollo de esta interfaz.

Buscar información acerca de la interfaz software implementada en la USRP.

9. AGRADECIMIENTOS

Al Instituto Geofísico del Perú por darme la oportunidad de desarrollar mi tesis para optar por el título de Ingeniero Electrónico en el Radio Observatorio de Jicamarca.

A mis compañeros: Ing. John Kerim Rojas Paredes, Ing. Juan Carlos Espinoza e Ing. Joaquín Verástegui por su guía, paciencia y enseñanzas en el campo de la ingeniería.

10. BIBLIOGRAFÍA

Estructura del USRP: https://kb.ettus.com/UHD#GNU_Radio0

GNU Radio: <https://www.gnuradio.org/about/>

Pasos para la creación de un módulo (Out of three)

<https://wiki.gnuradio.org/index.php/OutOfTreeModules>

Jicamarca raw and process data structure. Autor: Radio Observatorio de Jicamarca.

JARS 1.0: Radio Observatorio de Jicamarca.(2010). Development of the new Jicamarca Acquisition Radar System: JARS. (JARS 1.0). Sitio Web: <https://repositorio.igp.gob.pe/handle/IGP/3443>.

JARS 2.0: IEEE Xplore Digital Library. (2017). Design and implementation of a high speed interface system over Gigabit Ethernet based on FPGA for use on radar acquisition systems. (JARS 2.0). Sitio Web: <https://ieeexplore.ieee.org/document/8247311>.

DETECCIÓN DE ECOS IONOSFÉRICOS EN IONOGRAMAS DIGITALES USANDO REDES NEURONALES CONVOLUCIONALES

**Ionospheric Echoes Detection in Digital Ionograms Using Convolutional Neural Networks*

César De La Jara Sánchez

Pontificia Universidad Católica del Perú
Escuela de posgrado – Maestría en Informática
cesar.delajara@jro.igp.gop.pe

Asesor: César Olivares Poggi

Profesor de Maestría en Informática de la PUCP

RESUMEN

Un ionograma es un gráfico que muestra la distancia que una onda transmitida verticalmente, de una frecuencia determinada, viaja antes de regresar a la tierra. El ionograma se forma haciendo un trazo de esta distancia, denominada altura virtual, contra la frecuencia de la onda transmitida. Junto con los ecos de la ionósfera, los ionogramas suelen contener una gran cantidad de ruido de diferente naturaleza, que debe ser eliminado con el fin de extraer información útil. En el presente trabajo se propone utilizar un modelo de red neuronal convolucional para mejorar la calidad de la información obtenida a partir de ionogramas digitales, en comparación con la del uso de técnicas de procesamiento de imágenes y aprendizaje automático, en la generación de perfiles de densidades electrónicas. Un conjunto de datos de más de 900.000 ionogramas de 5 estaciones de observación ionosférica está disponible para su uso.

ABSTRACT

An ionogram is a graph that shows the distance that a vertically transmitted wave, of a given frequency, travels before returning to the earth. The ionogram is shaped by making a trace of this distance, which is called virtual height, versus the frequency of the transmitted wave. Along with the echoes of the ionosphere, ionograms usually contain a large amount of noise from different sources, that must be removed in order to extract useful information. In the present work, we propose the use a convolutional neural network model to improve the quality of the information obtained from digital ionograms, compared to that using image processing and machine learning techniques, in the generation of electronic density profiles. A data set of more than 900,000 ionograms from 5 ionospheric observation stations is available to use.

Keywords: Ionosphere, ionosondes, ionograms, deep learning, convolutional neural network, ionogram inversion.

1. INTRODUCCIÓN

Las ionosondas son un tipo de radar que envía pulsos verticales de ondas de radio de alta frecuencia a la ionósfera. Los ecos de estos pulsos se registran en la Tierra y se utilizan para generar un tipo de trazos representativos de la ionósfera, llamados ionogramas. Existe una relación directa entre la frecuencia de los pulsos transmitidos y las densidades de ionización de las capas ionosféricas que las reflejan [1].

Los ecos forman patrones característicos que conforman un ionograma. La velocidad de los pulsos que viajan en la ionósfera es menor que la velocidad de los pulsos que viajan en el espacio libre, por lo tanto, después de un viaje completo, es posible obtener información sobre la altura aparente o la altura virtual de las capas. La altura virtual de los pulsos cuyas frecuencias se aproximan a la frecuencia de plasma máxima de una capa dada tiende al infinito, porque los pulsos deben viajar una distancia finita a una velocidad efectiva cercana a cero. Las frecuencias a las que se produce este fenómeno se denominan frecuencias críticas [2]. A partir de los ionogramas es posible escalar manualmente o mediante métodos computacionales valores característicos de alturas virtuales $h'E$, $h'F$, $h'F2$, etc. y frecuencias críticas foE , $foF1$, $foF2$, etc. de cada capa de la ionósfera [3].

El interés en escalar e interpretar ionogramas está aumentando entre la comunidad científica, la dificultad de estos procesos es que extraer parámetros ionosféricos a través del escalado manual es una tarea muy exigente tanto en esfuerzo como en tiempo. Con el desarrollo de las técnicas de procesamiento de imágenes, se han realizado avances en el escalado automático de ionogramas [4] [5], en el que se hace hincapié principalmente en el reconocimiento de las trazas de ionogramas, luego se obtienen perfiles de densidad aplicando técnicas de inversión ionosférica. En situaciones de alta complejidad, como en el caso de los ionogramas incompletos o con la presencia de fenómenos ionosféricos como la F dispersa, el escalado automático se vuelve más difícil y tiende a fallar [6].

En el presente trabajo proponemos el uso de un modelo de red neuronal convolucional para la detección de ecos de ionósfera en ionogramas digitales, que puede servir como una herramienta para el escalado automático.

El trabajo está estructurado de la siguiente manera: la Sección 2 ofrece una breve introducción al modo de operación de las ionosondas, así como a las características de la ionósfera. La Sección 3 describe la métrica de rendimiento que se utilizará. La Sección 4 describe y evalúa el desempeño de los modelos de referencia. La Sección 5 describe el uso de redes neuronales convolucionales para la detección de perfiles.

2. LA IONÓSFERA

Los ionogramas son diagramas generados por las ionosondas, que muestran el tiempo que tarda una onda de cierta frecuencia en ser reflejada por la ionósfera. Un ionograma es una representación del estado de la ionósfera en un momento dado.

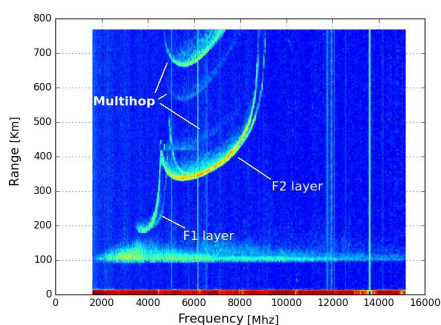


Fig. 1. Ionograma de la ionosonda de Jicamarca que muestra la altura ionosférica virtual (vertical) versus la frecuencia de los pulsos transmitidos (horizontal). Las trazas muestran las capas F1 y F2, así como el salto múltiple.

La figura 1 muestra un ejemplo de un ionograma típico. El eje horizontal representa las frecuencias de las señales de radio transmitidas y el eje vertical las alturas virtuales de las capas ionosféricas. El color de la traza es proporcional a la intensidad de la señal recibida, los ecos más intensos tendrán colores más cercanos al rojo, mientras que los ecos más débiles al azul. En esta imagen también se pueden distinguir claramente las capas F1 en una altura de 200 km y F2 en más de 300 km.

3. MEDIDA DE CALIDAD

En este trabajo proponemos un método para extraer capas ionosféricas de ionogramas digitales mediante la segmentación de imágenes con redes neuronales convolucionales, donde el objetivo principal es etiquetar cada píxel de la imagen como perteneciente al trazo ionosférico o al fondo.

La medida de calidad más usada para los problemas de segmentación de objetos es un índice llamado intersección sobre unión (IoU) [11]. IoU proporciona una relación entre el número de píxeles comunes en dos imágenes y el número total de píxeles en ambas imágenes. Si las imágenes son exactamente iguales, el resultado de esta relación sería 1, si hubiera pocas coincidencias entre las imágenes (imágenes muy diferentes), el resultado sería cercano a 0.

$$IoU = \frac{areacomún}{areatotal} \quad (1)$$

Como estamos trabajando con matrices binarias, el área se obtiene haciendo una suma de todas las celdas de la matriz. La intersección se obtendrá realizando una operación AND lógica entre los bits de las matrices y la unión se obtendrá realizando un OR lógico.

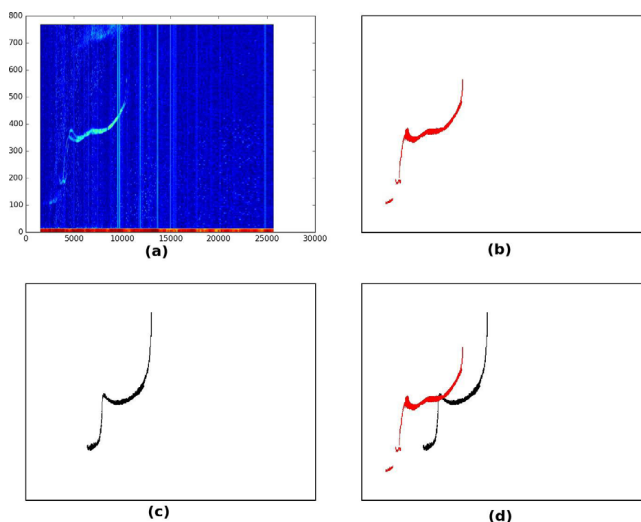


Fig. 2. (a) ionograma original, (b) segmentación manual, (c) segmentación automática, (d) comparación entre segmentación manual y automática.

La Figura 2 muestra la comparación de una segmentación manual (rojo) con una segmentación automática (negro). En este caso, a nivel de bit, la matriz de intersección tendrá solo unas pocas celdas en 1, lo que dará como resultado una IoU cercana a cero.

4. IMPLEMENTACIÓN Y EVALUACIÓN DE MODELOS DE LÍNEA BASE

A. Detección de perfiles usando procesamiento de imágenes y umbralización.

En este enfoque los ionogramas se consideran imágenes en las que el ruido debe filtrarse para aislar los ecos ionosféricos. Para este propósito se utilizan tres tipos de filtrado, un filtro median, el filtro definido por la matriz de Ker (2) y la umbralización.

$$K_{er} = \begin{bmatrix} 1110111 \\ 1110111 \\ 1110111 \\ 1110111 \\ 1110111 \\ 1110111 \end{bmatrix} \quad (2)$$

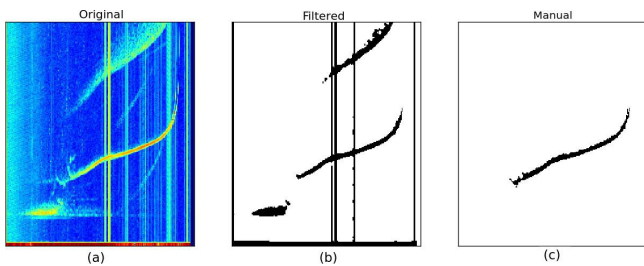


Fig. 3. Comparación entre el ionograma original (a), el resultado de la aplicación en cascada del filtro Ker, el filtro mediano y el umbral (b) y un ionograma segmentado manualmente (c)

Como se muestra en la figura 3, es posible eliminar con precisión el ruido de fondo, pero no se pueden eliminar la interferencia, ni la señal de calibración ni el multi hop. La IoU promedio entre los ionogramas del conjunto de prueba donde se aplicaron los tres filtros y los ionogramas segmentados manualmente es 0.163.

B. Detección de perfiles usando modelos de aprendizaje de máquina no supervisada

El otro método se basa en la representación de los ionogramas en matrices tridimensionales x, y, V donde x e y representan coordenadas espaciales y V la intensidad del punto. Con esta representación se aplican 2 técnicas de agrupamiento no supervisado, K-Means y Mean Shift.

La agrupación en clúster es una técnica para encontrar dentro de un conjunto de muestras, grupos que tienen características similares, por lo que las muestras que comparten características comparables pertenecerán al mismo grupo y se separarán de otros grupos. El objetivo es maximizar las variaciones entre grupos y minimizar las variaciones dentro de los grupos [12]. La Figura 3 muestra una comparación entre un ionograma original, un ionograma segmentado usando el algoritmo K-means y un ionograma segmentado manualmente.

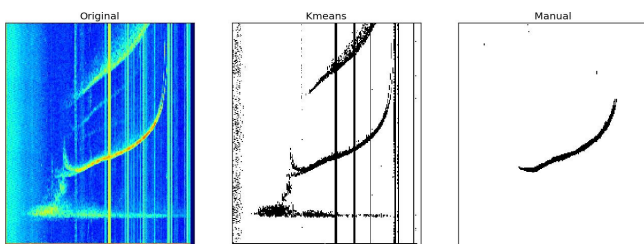


Fig. 4. Segmentación de ionogramas usando K means

	Filtered	K-Means	Mean Shift
IoU	0.163	0.157	0.105

Tabla 1. IOU PROMEDIO DE LOS MODELOS DE LÍNEA BASE

La IoU promedio entre los ionogramas segmentados usando el algoritmo K-means y los ionogramas segmentados manualmente para el conjunto de prueba es 0.157.

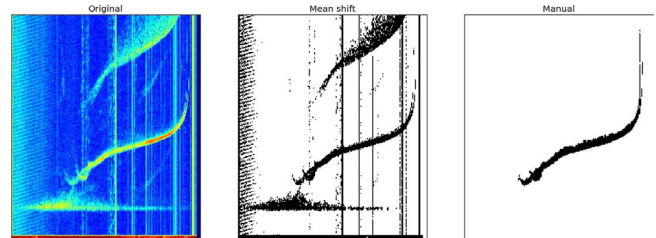


Fig. 5. Segmentación de ionogramas con mean shift

La IoU promedio entre los ionogramas segmentados usando mean shift y los ionogramas segmentados manualmente para el conjunto de prueba es 0.105.

La Tabla I resume los resultados de la IoU para los modelos de línea base.

5. DETECCIÓN DE PERFILES USANDO REDES NEURONALES CONVOLUCIONALES

Como se mencionó anteriormente, tenemos disponible una gran cantidad de datos no etiquetados en la base de datos LISN, por otro lado, los datos etiquetados solo se pueden obtener mediante la segmentación manual de ionogramas, que no solo es un proceso lento, sino que también requiere un alto nivel de conocimiento y experiencia. Dado este escenario, con el objetivo de construir un modelo de aprendizaje semi supervisado, se han utilizado ambos tipos de datos, la gran cantidad de datos no etiquetados y el pequeño número de ionogramas etiquetados (segmentados manualmente).

Utilizamos un modelo basado en una red neuronal convolucional con múltiples capas de convolución, de tipo decodificador codificador. La primera parte de la red es un codificador que asigna entradas crudas a una representación de vectores de características, la segunda parte es un decodificador que toma estas representaciones de características como entrada, las procesa, produce una salida y asigna la salida nuevamente al formato crudo. El objetivo de la red es aprender una representación eficiente de los datos.

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	(None, 256, 208, 1)	0
conv2d_15 (Conv2D)	(None, 256, 208, 16)	160
max_pooling2d_7 (MaxPooling2)	(None, 128, 104, 16)	0
conv2d_16 (Conv2D)	(None, 128, 104, 8)	1160
max_pooling2d_8 (MaxPooling2)	(None, 64, 52, 8)	0
conv2d_17 (Conv2D)	(None, 64, 52, 8)	584
max_pooling2d_9 (MaxPooling2)	(None, 32, 26, 8)	0
conv2d_18 (Conv2D)	(None, 32, 26, 8)	584
up_sampling2d_7 (UpSampling2)	(None, 64, 52, 8)	0
conv2d_19 (Conv2D)	(None, 64, 52, 8)	584
up_sampling2d_8 (UpSampling2)	(None, 128, 104, 8)	0
conv2d_20 (Conv2D)	(None, 128, 104, 16)	1168
up_sampling2d_9 (UpSampling2)	(None, 256, 208, 16)	0
conv2d_21 (Conv2D)	(None, 256, 208, 1)	145
Total params: 4,385		
Trainable params: 4,385		
Non-trainable params: 0		

Fig. 6. Modelo de red neuronal

La figura 6 muestra una descripción del modelo de red. Utilizamos seis capas de convolución con activación ReLU, 3 max pooling con cero padding y 3 up samplings. Se usa binary cross entropy como función de pérdida y adadelta como optimizador.

La figura 7 muestra detalles de las distintas capas del modelo.

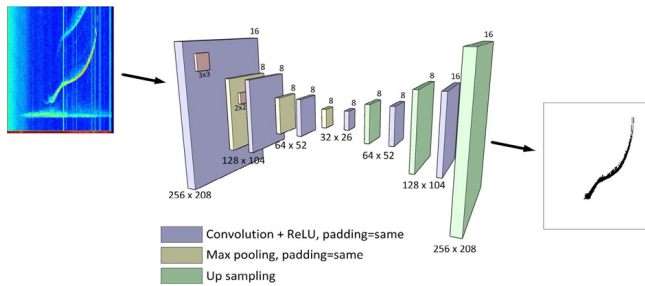


Fig. 7. Capas del modelo de red neuronal

En la primera etapa del proceso de aprendizaje, alimentamos la red neuronal con ionogramas originales (X) de los conjuntos de entrenamiento / validación, y usamos como variables de salida (y) ionogramas que se han segmentado utilizando las tres técnicas de referencia (filtros, k-medias y cambio medio). Terminamos con tres modelos de redes neuronales que aprendieron a reconstruir ionogramas de la misma manera que lo hacen los modelos de línea de base, como se muestra en la figura 8.

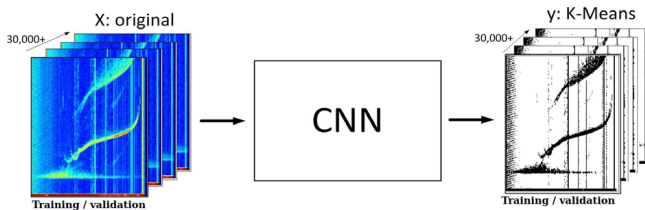


Fig. 8. X = ionogramas originales, y = ionogramas segmentados con K-means

En la figura 9 se muestra una comparación entre la aplicación de estos modelos en el conjunto de prueba con ionogramas segmentados manualmente.

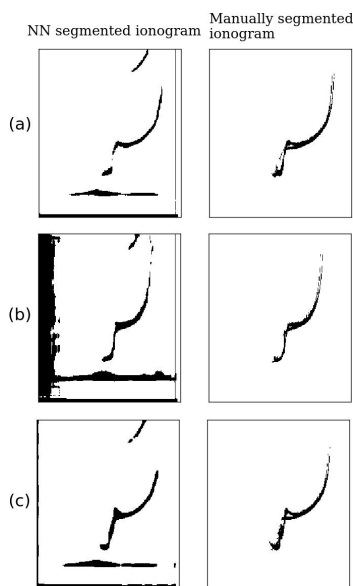


Fig. 9. Aprendizaje con modelos de línea base. La columna de la izquierda es la reconstrucción de un ionograma utilizando un modelo de red neuronal entrenado para reconstruir un ionograma segmentado con un modelo de línea base, y la columna derecha es la segmentación manual del ionograma. (a) K-means, (b) Mean shift (c) filtros

	Filtered	K-Means	Mean Shift
IoU	0.174	0.173	0.077

Tabla 2. IOU PROMEDIO DE LA RED NEURONAL

	Filtered	K-Means	Mean Shift
IoU	0.589	0.602	0.593

Tabla 3. IOU PROMEDIO DE LA RED CONVOLUCIONAL CON SINTONÍA FINA

El conjunto de entrenamiento utilizado para entrenar el modelo de red neuronal tiene 31,699 ionogramas, el conjunto de validación 9,577 y el conjunto de prueba 9,505. Se escalaron manualmente 816 ionogramas, de ellos 474 se usaron como conjunto de entrenamiento, 151 como conjunto de validación y 191 como conjunto de prueba. La IoU promedio entre los ionogramas del conjunto de prueba segmentados con los tres modelos y los ionogramas segmentados manualmente se resumen en la tabla II. Se observa una pequeña mejora en comparación con la IoU de los modelos de referencia.

En la segunda etapa del proceso de aprendizaje, sintonizamos finamente los modelos previamente entrenados. Los 3 modelos se alimentan nuevamente con ionogramas originales de los conjuntos de entrenamiento / validación, pero las variables de salida son esta vez ionogramas segmentados manualmente de los conjuntos de entrenamiento/validación.

La figura 10 muestra los resultados de las diferentes etapas del proceso de entrenamiento. De izquierda a derecha: el ionograma original, ionograma segmentado utilizando modelos de referencia, ionograma escalado manualmente, segmentación después de un ajuste fino. En este último se observa que la interferencia, el ruido de fondo, la señal de calibración del radar y la capa E se han eliminado por completo.

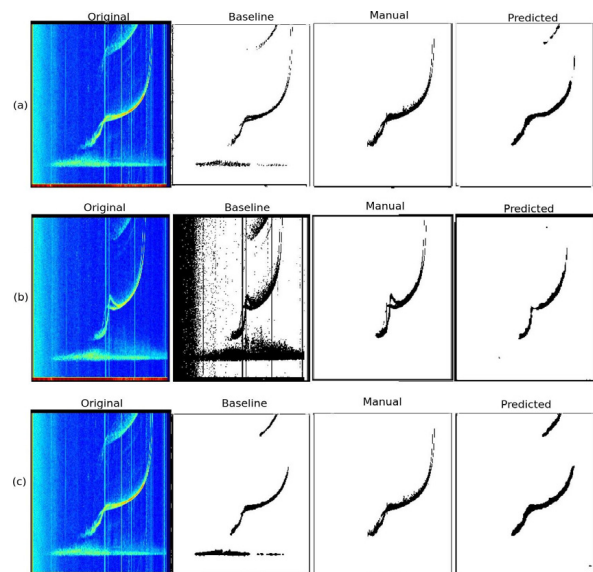


Fig. 10. Sintonía fina y predicción final (a)K means, (b)Mean shift (c) filters

La IoU promedio entre los ionogramas del conjunto de prueba segmentados con modelos con sintonía fina y los ionogramas segmentados manualmente se resume en la tabla III. Se observa una mejora significativa en comparación con la IoU de los ionogramas segmentados antes de aplicar el ajuste fino.

Como etapa final, el modelo CNN fue entrenado usando solo ionogramas segmentados manualmente como datos de salida, lo que dio un IoU promedio de 0.569.

6. CONCLUSIONES

En términos de IoU, la segmentación de ionogramas cuando se usa una red neuronal entrenada usando datos etiquetados es 5.6 veces mejor que la segmentación realizada por los modelos de línea base.

El uso de grandes cantidades de datos no etiquetados para generar un modelo previamente entrenado mejora ligeramente (<6%) la precisión del modelo final, por lo que podemos decir que el rendimiento del modelo se basa principalmente en el uso de una red neuronal convolucional y no en un proceso de pre-entrenamiento con datos no etiquetados.

7. REFERENCIAS

- [1] W. Piggot and K. Rawer, *URSI Handbook of Ionogram Interpretation and Reduction*. World Data Center for Solar-Terrestrial Physics, 1972.
- [2] A. S. Jursa, *Handbook of Geophysics and the Space Environment*. Air Force Geophysics Lab Hanscom AFB MA, 1985.
- [3] L. Perrone, A. Mikhailov, C. Cesaroni, L. Alfonsi, A. De Santis, P. Michael, and C. Scotto, "Long-term variations of the upper atmosphere parameters on rome ionosonde observations and their interpretation," *J. Space Weather Space Clim*, 2017.
- [4] C. Scotto and M. Pezzopane, "A software for automatic scaling of fof2 and muf(3000) f2 from ionograms," *URSI XXVIIth General Assembly*, 2002.
- [5] C. Ziwei, W. Shun, Z. Shunrong, F. Guangyou, and W. Jinsong, "Automatic scaling of f layer from ionograms," in *Radio Science*, vol. 48, 2013, pp. 334–343.
- [6] S. M. Stankov, J.-C. Jodogne, I. Kutiev, K. Stegen, and R. Warnant, "Evaluation of automatic ionogram scaling for use in real-time ionospheric density profile specification: Dourbes dgs-256/artist-4 performance," *Annals of Geophysics*, 2012.
- [7] "observatorio distribuido lisn". Instituto Geofísico del Peru, 2019. Accessed: 2019-02-15. [Online]. Available: <http://lisn.igp.gob.pe>
- [8] "lisn data user rules". Instituto Geofísico del Peru, 2019. Accessed: 2019-02-15. [Online]. Available: <http://lisn.igp.gob.pe/data/>
- [9] T. Bullet. (2015) Software for reading VIPIR raw data files. National Oceanics and Atmospheric Administration. Accessed: 2019-01-15. [Online]. Available: <ftp://ftp.ngdc.noaa.gov/>
- [10] "network common data form (netcdf)". University Corporation for Atmospheric Research. Accessed: 2019-01-15. [Online]. Available: <https://www.unidata.ucar.edu/software/netcdf/>
- [11] M. A. Rahman and Y. Wang, "Optimizing intersection-over-union in deep neural networks for image segmentation," in *International Symposium on Visual Computing*. Springer, 2016, pp. 234–244.
- [12] D. Pham, S. Dimov, and C. Nguyen, "Selection of k in k-means clustering," *Journal of Mechanical Engineering Science* 1989-1996 (vols 203-210), 2005.
- [13] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol 17 no 8, 1995.

DISEÑO DE UN SISTEMA DE DETECCIÓN TEMPRANA DE HUAICOS UTILIZANDO PROCESAMIENTO DE IMÁGENES

**Design of a huaico early detection system using image processing*

Juan Carlos Barbaran Meza

Universidad de Ingeniería y Tecnología
Facultad de Ingeniería a Electrónica
Juan.barbaran@jro.igp.gob.pe

Asesor: Joaquín Verástegui

Radio Observatorio de Jicamarca - IGP

RESUMEN

En el presente trabajo se expone el diseño de un sistema para la detección de huaicos utilizando procesamiento de imágenes adquiridas por una cámara IP. La determinación de la presencia de huaicos se realizó mediante la comparación de dos imágenes adquiridas de una manera correlativa. El principal algoritmo utilizado para determinar la presencia de huaicos fue "Structural Similarity" (SSIM) el cuál detecta diferencias estructurales no captadas por simple diferenciación, pero sí por el ojo humano. Este algoritmo combinado con otros métodos de procesamiento de imágenes logró la detección de huaicos en los eventos de marzo del 2017. Así mismo se logró diferenciar entre huaicos y riachuelos que pueden aparecer en el mismo cauce. La explicación de la metodología utilizada y los resultados obtenidos se detallan en las siguientes páginas.

Palabras clave: Detección de Huaicos, Procesamiento de imágenes, Similitud estructural

ABSTRACT

In the present work the design of a system for the detection of huaicos using image processing acquired by an IP camera is presented. The determination of the presence of huaicos was made by comparing two acquired images in a correlative way. The main algorithm used to determine the presence of huaicos was "Structural Similarity" (SSIM) which detects structural differences not captured by simple differentiation. This algorithm combined with other methods of image processing achieved the detection of huaicos during the event of March 2017. Likewise, it was possible to differentiate between huaicos and riachuelos that may appear in the same channel. The explanation of the methodology used and the results obtained are detailed in the following pages.

Keywords: Huaico detection, image processing, structural similarity

1. INTRODUCCIÓN

Durante el primer semestre del 2017 el Perú sufrió los estragos del denominado Niño Costero, el cual produjo que las aguas del Pacífico eleven sus temperaturas y se registren fuertes lluvias. Estas a su vez ocasionaron que los principales ríos se desborden de su cauce original, provocando pérdidas humanas y económicas en gran parte de la costa de nuestro país. Según el Centro de Operaciones de Emergencia Nacional (COEN), solo en Lima, se registraron 9,550 damnificados por las inundaciones que dejaron los huaicos producidos a las afueras de la capital.

En las afueras de Lima se ubican los ríos Huaycoloro y Río Seco, estos se encuentran en los distritos de Lurigancho - Chosica y Huachipa, y durante el desastre del año pasado fueron las zonas más afectadas de la capital. Por tal motivo, el Radio Observatorio de Jicamarca (ROJ) decidió implementar un sistema monitoreo de huaicos utilizando una cámara ubicada en las proximidades del cauce del río Huaycoloro.

Este trabajo tiene como finalidad explicar el proyecto realizado durante el programa de prácticas del presente año, por tal motivo se ha considerado presentar inicialmente una lista de conceptos claves que serán de ayuda para poder entender con mayor facilidad todos los pasos que siguieron para su realización.

2. DESARROLLO

2.1. Marco teórico

2.1.1. Características de las imágenes

Las características de las imágenes que se usarán para el procesamiento son fundamentales, puesto que sirve para poder entender el problema y aplicar los mejores métodos para lograr el objetivo. La cámara utilizada para la obtención de dichos gráficos es una del tipo IP que realiza capturas a una resolución de 720 x 1280 píxeles, así mismo estas son enviadas al servidor cada 10 segundos.

Por otro lado, la metodología adoptada para implementar el sistema de detección es la comparación entre dos imágenes capturadas de manera consecutiva, por ello es importante identificar los problemas que pueden presentarse al comparar dos imágenes. Los principales problemas encontrados son dos:

Movimiento de la cámara: Debido al viento, la cámara sufre ligeros movimientos provocando que la captura de imágenes no se realice con el mismo enfoque, dificultando su comparación.

Luminosidad: Debido al movimiento de las nubes y otros factores ambientales, las fotos capturadas presentan diferente luminosidad.

El objetivo inicial para implementar el sistema de detección de huaicos será eliminar el ruido y error provocado por la luminosidad y el movimiento de la cámara para una correcta identificación.

2.1.2. Similitud Estructural (SSIM)

Este método fue publicado en abril del 2014 en el "IEEE Transactions on Image Processing" por Wang y Bovik. A diferencia de los métodos tradicionales de comparación, en los cuales se calcula el error absoluto. En SSIM se tiene en cuenta variables como la luminosidad, contraste y estructura. Estos parámetros son percibidos por el ojo humano fácilmente. El valor de SSIM varía entre -1 a 1, siendo 1 el valor que toma si las imágenes son idénticas.

2.1.3. Coeficiente de Correlación mejorado (ECC)

El registro de imágenes es una técnica que se utiliza para poder normalizar imágenes que están desalineadas entre sí. Para ello se debe estimar un modelo de movimiento, el cual puede ser de diferentes tipos, sin embargo, en este trabajo se utilizaron dos tipos: Traslación y Rotación.

El ECC es un método de registro de imágenes propuesto por Georgios D. Evangelidis que, a diferencia de otros métodos, éste no es afectado por cambios fotométricos en contraste o luminosidad, esta última característica hace que sea el adecuado para este trabajo.

2.1.4. Características de Haralick

Estas características propuestas por Haralick utilizan medidas estadísticas de la "Matriz de co-ocurrencia", la cual es una tabulación de cuántas veces diferentes combinaciones de tonalidades de grises pueden aparecer en una imagen. Haralick propuso 14 características las cuales son:

1. Segundo momento Angular
2. Contraste
3. Correlación
4. Varianza
5. Momento diferencial inverso
6. Suma de promedios
7. Suma de varianzas
8. Suma de entropías
9. Entropía
10. Varianza diferencial
11. Entropía Diferencial
12. Medición de correlación 1
13. Medición de correlación 2
14. Máximo coeficiente de correlación

Estas características servirán para diferenciar un huaico de un riachuelo, ya que poseen diferentes texturas.

2.2. Metodología

2.2.5. Configuración inicial

Debido a que es conocido el cauce por donde pasará el huaico, se realizará una configuración manual en la que el usuario seleccionará la zona que será monitoreada por el sistema y se generará una máscara, limitando así la zona de análisis. Esta práctica es común en el procesamiento de imágenes ya que elimina ruido o errores causados por distorsiones en zonas ajenas a la que se quiere analizar.

2.2.6. Pre procesamiento

Antes que las imágenes sean comparadas, se realizará un pre-procesamiento el cual consiste de tres etapas:

Cortar la imagen: En esta etapa, se corta la imagen para eliminar las letras de fecha y hora.

Pasar a escala de grises: Todo el procesamiento de imágenes en este sistema se realizará con imágenes en escala de grises por ello se debe convertir la imagen de color a una en escala de grises.

Ecuilibración de histograma: Por último, se ecualiza el histograma para aumentar el contraste y eliminar las diferencias de luminosidad entre imágenes.

2.2.7. Extracción de características

La última etapa es la más larga debido a que es el núcleo de todo el sistema. En la *figura 1* se muestra un esquema general de la extracción de características.

Después de que las dos imágenes a comparar han pasado el pre procesamiento, se procede al alineamiento de la "imagen actual" (B) utilizando ECC, obteniéndose una imagen alineada a la "imagen previa" (A). Seguidamente, la imagen A y la imagen

B alineada son enmascaradas utilizando la máscara creada en la etapa de configuración.

Las imágenes enmascaradas son comparadas usando SSIM y se obtiene la primera característica: El "score promedio de similitud", así mismo se obtiene una matriz de scores individuales del SSIM en cada pixel, esta matriz de scores se normaliza de 0 a 255 para formar una imagen en escala de grises. En la *figura 2* se muestra un ejemplo del resultado.

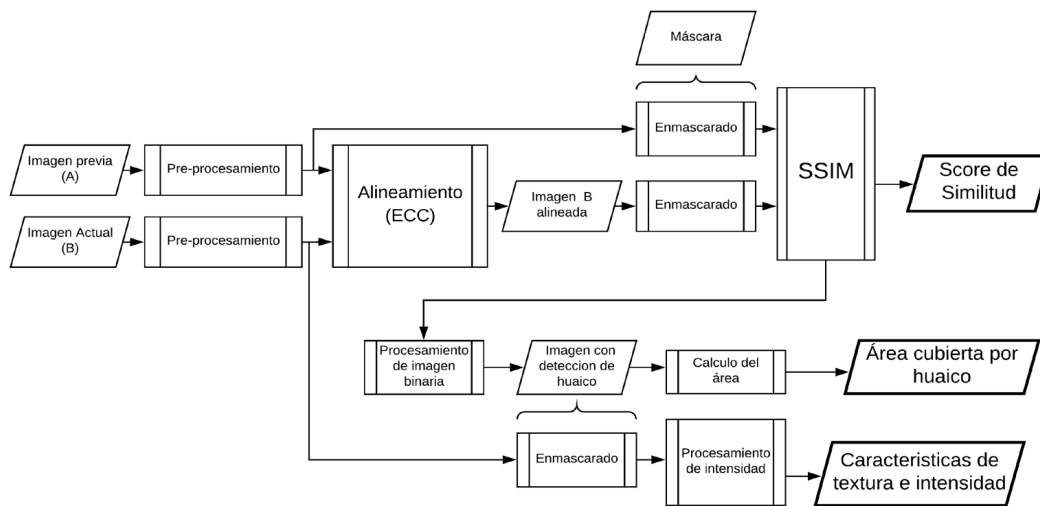


Figura 1: Diagrama de implementación de la extracción de características

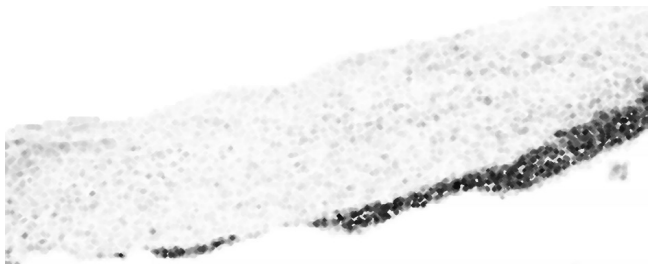


Figura 2: Imagen formada por los scores individuales de SSIM

Como se puede observar, se tiene zonas más oscuras (valores cercanos a 0) en donde se encuentran cambios drásticos entre las imágenes en este caso el pequeño riachuelo en la parte inferior del cauce del río.

Esta nueva imagen es procesada binariamente, es decir se le realizó un "threshold". Después de ello se realizó transformaciones morfológicas (dilataciones y erosiones) con el objetivo de obtener un bloque sólido que representen movimiento de agua y/o huaicos y a su vez eliminar ruido. El resultado se observa en la *figura 3*.



Figura 3: Identificación del riachuelo utilizando procesamiento binario

A esta imagen con procesamiento binario se le calculó el área (número de píxeles de color blanco) y se comparó con la zona sombreada en la configuración inicial para obtener la segunda característica: "El porcentaje de cambio en la región a analizar".

Finalmente, para la identificación de texturas primero se limitó el área de análisis a la zona identificada con movimiento, para ello se enmascara la imagen B con la imagen con procesamiento binario. Un ejemplo de éste paso se puede observar en la *figura 4*.

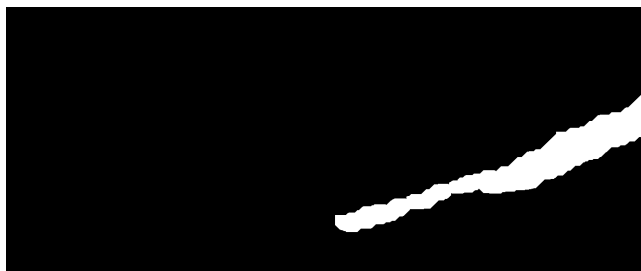


Figura 4: Imagen enmascarada para identificación de texturas

Después de enmascarar las imágenes se extrajeron las dos últimas características: "Características de Haralick" y la "Intensidad".

3. RESULTADOS

El sistema fue testeado en un set de 1340 imágenes pertenecientes al 16 de marzo del 2017. En esta base de datos, el huaico inicia a las 15:30 aproximadamente y previo a ello existe un riachuelo pequeño. El resultado final del sistema son las cuatro características, las cuales se describen a continuación:

Score promedio de similitud: Este valor indica qué tan similares son las imágenes, el valor se encuentra entre 0 y 100. En la *figura 5* se muestra los Score promedio de todas las imágenes de la base de datos ordenadas cronológicamente.

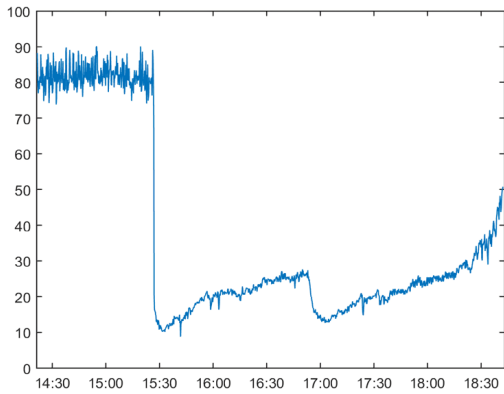


Figura 5: Scores de similitud de toda la base de datos

Área ocupada por la masa de agua: El área del cauce ocupada indica qué tan lleno está éste. El valor va de 0 a 100 %. En la figura 6 se muestra el área ocupada de todas las imágenes de la base de datos ordenada cronológicamente.

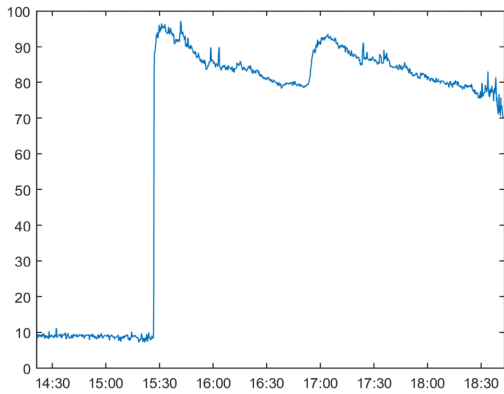


Figura 6: Área ocupada por la masa de agua de toda la base de datos

Características de Haralick: Después de un testeó, se determinó que las mejores características para este sistema son las características 1, 3 y 6. En las figuras 7, 8 y 9 se muestran las tres características de Haralick elegidas, cabe resaltar que éstas están normalizadas de 0 a 1.

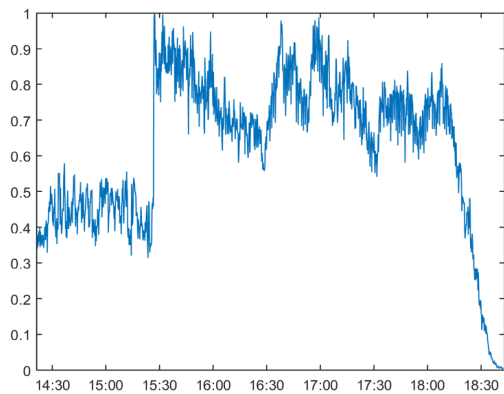


Figura 7: Característica 1 de Haralick de toda la base de datos

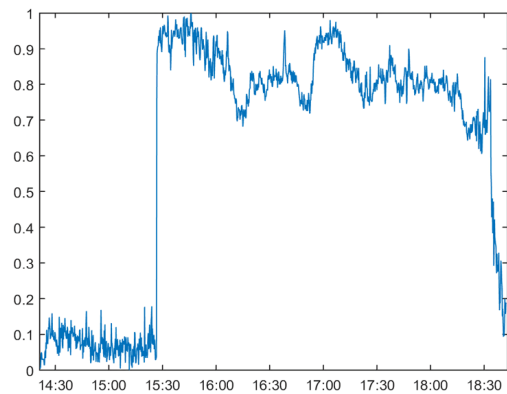


Figura 8: Característica 3 de Haralick de toda la base de datos

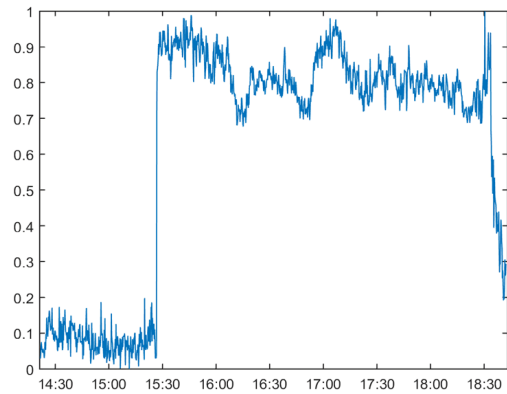


Figura 9: Característica 6 de Haralick de toda la base de datos

Intensidad promedio: Este valor varía de 1 a 255 debido a que esas son las intensidades en la escala de grises. En la figura 10 se muestra la intensidad promedio de todas las imágenes de la base de datos.

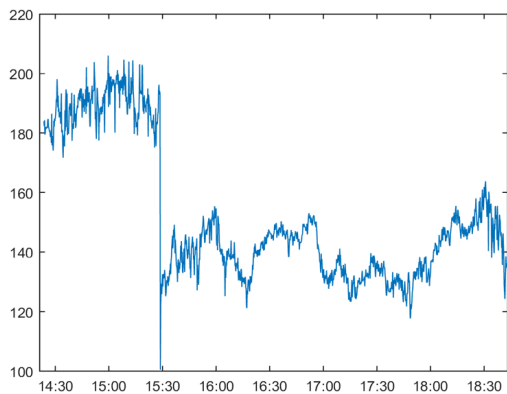


Figura 10: Intensidades de toda la base de datos

4. DISCUSIÓN

Score promedio de similitud: Se puede notar que hay una diferencia notoria a partir de las 15:30, es decir, el valor disminuye de un promedio de 80 a un promedio de 20, confirmando así que el sistema detecta un huaco desde ese punto.

Área ocupada por la masa de agua: Se puede notar que hay una diferencia notoria a partir las 15:30, de 10% pasa a un promedio de 85%. Cabe resaltar que al inicio se muestra el porcentaje de 10% debido a que en esta base de datos antes del

huaico existe un pequeño riachuelo.

Características de Haralick: De las 14 características presentadas por Haralick, tres fueron elegidas por mostrar mayor contraste ante la presencia de huaico y riachuelo. En la característica número 1, el cambio es menos aparente que en las características 2 y 3, sin embargo aún sigue siendo una buena fuente de información para determinar si el movimiento de agua es un riachuelo o un huaico.

Intensidad promedio: La intensidad de color es una característica que se aprecia a por simple inspección, debido a que un riachuelo tenderá a ser más claro (intensidad mayor) por su naturaleza uniforme de su curso; sin embargo un huaico tendrá sedimentos y lodo que harán que su tonalidad sea más oscura (intensidad menor), esto se comprueba en la imagen 10 donde se ve que la intensidad es menor cuando se presenta el huaico.

En casi todas las gráficas de los resultados se puede observar que después de las 18:00 horas las características no muestran lo esperado o no son eficientes, esto es debido a que la noche empieza y no es posible analizar imágenes con tan poca iluminación.

5. CONCLUSIONES

El pre procesamiento de las imágenes ayudó a disminuir el error producido por la luminosidad y el movimiento de la cámara.

El sistema implementado es capaz de detectar la presencia de huaicos de manera precisa y eficaz utilizando la combinación de las 4 características halladas.

El sistema es capaz de discriminar entre riachuelos o corrientes de agua y huaicos, lo cual es importante debido a que éstos son un indicador de un próximo huaico.

El sistema puede operar mientras la iluminación sea adecuada, en el caso de la cámara actual, es bastante difícil analizar las imágenes en la noche.

6. RECOMENDACIONES

Es posible que el sistema funcione de noche si se implementa una cámara con visión nocturna o si se ilumina la zona durante la noche.

Se debe probar el sistema con más muestras de fotos de riachuelos y huaicos para estandarizar las características de éstas y sea más fácil su identificación.

7. AGRADECIMIENTOS

Agradezco al Instituto Geofísico del Perú, a través del Radio Observatorio de Jicamarca, por haberme dado la oportunidad de pertenecer a su programa de Prácticas Pre-Profesionales, y mi más sincero agradecimiento al Ing. Joaquín Verástegui por su apoyo y asesoramiento.

8. BIBLIOGRAFÍA

Centro de Operaciones de Emergencia Nacional - COEN. (2017). Boletín informativo de emergencias. Reporte no. 79. Recuperado de: <https://www.mindef.gob.pe/informacion/documentos/%20BOLETINPRIMEROSETENTAYNUEVE.pdf>

Centro de Operaciones de Emergencia Nacional - COEN. (2017). Boletín informativo de emergencias. Reporte no. 138. Recuperado de: <https://www.mindef.gob.pe/informacion/documentos/BOLETIN%20INFORMATIVO%20N138.pdf>

Evangelidis, G & Psarakis, (208). "Parametric Image Alignment Using Enhanced Correlation Coefficient Maximization". Proc. IEEE Transactions on pattern analysis and machine intelligence,

vol. 30, no 10. Recuperado de: http://xanthippi.ceid.upatras.gr/people/evangelidis/george_files/PAMI_2008.pdf

González, R. & Woods, R. (2008). Digital Image Processing. New Jersey: Prentice Hall

Haralick, R. (1979). "Statistical and structural approaches to texture". Proc. IEEE, vol. 67, no. 5, pp. 786-804.

Kumar, M. Digital image processing. Photogrammetry and Remote Sensing Division Indian Institute of Remote Sensing. Recuperado de : <http://www.wamis.org/agm/pubs/agm8/Paper-5.pdf>

Organización Panamericana de la salud. (2017). Lluvias e inundaciones en el Perú. (no. 13). Recuperado de: http://www.paho.org/disasters/index.php?option=com_docman&view=download&category_slug=floods-peru-march-2017-227&alias=2513-ops-oms-reporte-situacion-13-23-abril-2017-3&Itemid=1179&lang=en

Red Humanitaria Nacional. (2017). Perú, temporada de lluvias (no. 4). Recuperado de: https://reliefweb.int/sites/reliefweb.int/files/resources/Redhum_PE-Sitrep_N4_inudaciones_peru_UN_31marzo2017-20170331-IA-20421.pdf

Wang, Zhou; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. (2004). "Image quality assessment: from error visibility to structural similarity". IEEE Transactions on Image Processing. 13 (4): 600-612.

DISEÑO DE UNA ARQUITECTURA DE SOFTWARE PARA PROCESAMIENTO PARALELO ESCALABLE EN SIGNAL CHAIN

**Design of a software framework capable of supporting scalable parallel processing in signal chain*

Sergio Cortez Gallegos

Universidad de Ingeniería y Tecnología
Departamento de Ingeniería Electrónica
sergio.cortez@jro.igp.gob.pe

Asesor: Ing. Juan Carlos Espinoza

Radio Observatorio de Jicamarca – IGP

RESUMEN

En el presente artículo se presenta una nueva arquitectura de software para la biblioteca de procesamiento Signal Chain del Radio Observatorio de Jicamarca. Esta tiene como objetivo paralelizar instrucciones relevantes a distintas unidades de procesamiento. Todo esto con el fin de utilizar de manera más eficiente los recursos disponibles, minimizar el tiempo de ejecución y realizar distintas tareas de manera simultánea. Un aspecto importante de la arquitectura presentada es su escalabilidad. Al estar descrita en un lenguaje de alto nivel como Python, se pueden integrar más unidades de procesamiento en otras plataformas de manera progresiva a medida que la capacidad de cómputo se intensifique. Diversos programas se probaron en esta nueva arquitectura resultando compatible con las unidades de procesamiento ya descritas y el flujo de la herramienta Signal Chain. Entre los resultados esperados, se prevé una reducción en los tiempos de ejecución y el uso total de la capacidad de procesamiento disponible.

Palabras clave: Signal Chain, paralelismo, multiproceso, escalabilidad

ABSTRACT

In the present article, a new software framework for the Signal Chain processing library of the Jicamarca Radio Observatory is presented. It aims to parallelize instructions relevant to multiple processing units. All this with the purpose of using the available resources in a more efficient way, reduce execution time, and achieve concurrency. An important aspect of this new framework is its scalability. As it is described in a high-level programming language, the integration of more computation platforms to leverage the workload can be accomplished. Different experiments have been tested using the enounced software architecture, being compatible with the flow of Signal Chain.

Keywords: Signal Chain, parallelism, multiprocessing, scalability

1. INTRODUCCIÓN

En las últimas décadas, la introducción de nuevas arquitecturas MIMD (*multiple instruction, multiple data*) en procesadores comerciales han generado la tendencia de emplear el paralelismo como un paradigma en el desarrollo de programas y aplicaciones. Esto con el fin de aprovechar los recursos disponibles al máximo, disminuir el tiempo de ejecución y ser fácilmente escalables. Supercomputadores como Blue Gene, de IBM, se basan en este tipo de arquitecturas para poder realizar cálculos que requieren una capacidad de cómputo inmensa en tiempos relativamente cortos.

1.1. Paralelismo y multiprocesamiento

Por definición, el cómputo paralelo plantea la capacidad de ejecutar diversas instrucciones de un programa al mismo tiempo. Esto considerando si son independientes entre sí y trabajan con un espacio de memoria definido. Dependiendo del nivel de abstracción en el cual se describa un programa, se puede especificar qué instrucciones deben realizarse de manera paralela (bajo nivel) o qué procesos se deben ejecutar de manera concurrente (alto nivel). Considere que la acción multiproceso en un procesador no refleja necesariamente paralelismo. Un núcleo puede ejecutar varios procesos con una determinada granularidad, pero no de manera concurrente.

1.2. Signal Chain

Signal Chain es la herramienta que emplea el Radio Observatorio de Jicamarca para procesar datos obtenidos de radares [1]. Este presenta una arquitectura modular; es decir, cada algoritmo de procesamiento es separado en 'bloques'. Estos bloques forman parte de una jerarquía bien definida la cual se tiene como referencia para describir algún experimento. La Fig. 1 ilustra las capas que posee Signal Chain. La capa de configuración describe la arquitectura general para el experimento ya sea por medio de un script o usando una GUI (interfaz gráfica de usuario). Aquí se define la cantidad de proyectos y las unidades de procesamiento y operaciones que estos poseen. La capa de proyecto (también llamado controlador) se encarga de instanciar ordenadamente las unidades y de asignar las operaciones correspondientes a cada una de ellas. Las unidades manipulan objetos que contienen data procesada y por procesar. Cada unidad tiene operaciones específicas asignadas que ejecutan algoritmos sobre los objetos en un orden determinado.

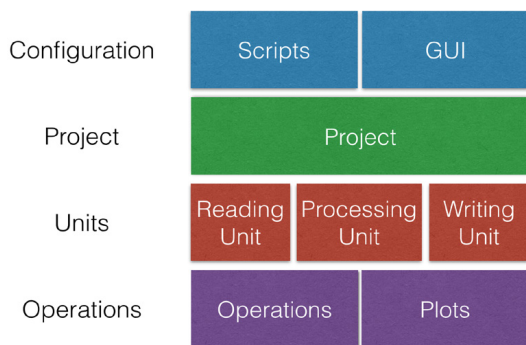


Figura 1: Capas jerarquizadas de la herramienta Signal Chain. Obtenido de [1].

El estado actual de la herramienta Signal Chain posee la característica de tener como subprocesos independientes a cada proyecto que se instancie en un script. Técnicamente, si se tuvieran dos proyectos descritos los cuales no dependen de entre sí, estos deberían ejecutarse en hilos (*threads*) separados de manera concurrente. Sin embargo, una ramificación en la estructura en una unidad de procesamiento no podría ejecutarse

de manera concurrente con la otra unidad en su misma jerarquía. Es la finalidad del proyecto aumentar la concurrencia dentro de un proyecto mediante una nueva arquitectura de software.

2. DESARROLLO

La arquitectura propuesta debe ser altamente escalable y capaz de soportar un grado de paralelismo de acuerdo con los posibles hilos que se describan con ella.

2.1. Procesos y subprocesos

Signal Chain posee unidades para el procesamiento de la data y operaciones que modifican la misma de manera definida. Nuestra nueva arquitectura pretende aumentar el grado de paralelismo dentro de un proyecto si este contiene bifurcaciones en el hilo inicial (o hilos iniciales, sería posible inicializar más de uno). Esto haría posible la ejecución de más de un hilo de manera concurrente dentro de un mismo proyecto. Técnicamente estamos reduciendo el nivel de abstracción en la arquitectura a fin de indicar al sistema operativo que instrucciones deberán ejecutarse de manera paralela si fuera posible. Por tal motivo, las unidades y operaciones tienen que tratarse como subprocesos.

2.1.1. Decorador de clases

El controlador (proyecto) en Signal Chain posee métodos para instanciar las unidades y operaciones descritas en la estructura del experimento. Todo el software está orientado a trabajar con objetos y por tal motivo cada unidad y operación tienen etapas de configuración antes de su instanciación. Modificar aquello para darles al grado de subproceso no es práctico y debido a la dependencia entre dichos objetos de configuración, se torna complicado cambiar su naturaleza. Se optó por usar un decorador para elevar de manera indirecta las unidades y operaciones al grado de subprocesos independientes.

2.2. Comunicación entre procesos (IPC)

La comunicación entre procesos en el caso que no exista memoria compartida y cada subproceso es ajeno en términos de variables con otros subprocesos es la más sencilla de implementar. Sockets, filas y tuberías son las soluciones más comunes.

2.2.1. Sockets

Un socket es una abstracción usada en programación para el intercambio de información (data) entre dos procesos [3]. De manera simplificada, un proceso apertura un 'orificio' (socket) al cual está pendiente o escuchando, y otro proceso se conecta por dicho orificio y es capaz de transmitir a través de él información de valor. ZeroMQ (ZMQ) es una librería ampliamente difundida que ofrece sockets capaces de llevar mensajes en diversos escenarios con modalidades especiales [2][4]. En esta sección analizaremos algunas topologías que se prestan para la IPC en nuestra nueva arquitectura.

Además de la clásica modalidad cliente servidor, ZMQ agrega entre sus patrones básicos de mensajería las siguientes modalidades [4]:

- **Request – Reply (solicitud y respuesta):** El cliente en este caso 'solicita' algo a un servidor y este 'responde'.
- **Publisher – Subscriber (publicador y suscriptor):** esta topología permite que varios clientes se 'suscriban' a un publicador (servidor). El servidor 'publica' con un determinado tópico un mensaje y los que están suscritos a él con el mismo tópico lo puede recibir.
- **Push – Pull (presionar y retirar):** en esta topología, ya no se habla de cliente y servidor sino de un productor y múltiples consumidores. El productor 'presiona' un mensaje el cual llega a un primer consumidor, el productor vuelve a

presionar un mensaje y este llega a un segundo consumidor y así sucesivamente.

2.2.2. Colas (Queues) y tuberías (Pipes)

Otra aproximación empleada para comunicar procesos es por medio de colas y tuberías. Las implementaciones de estas son de más bajo nivel y más sencillas y limitadas en su funcionamiento general. Una tubería (pipe) puede comunicar de manera unidireccional o bidireccional a solo dos procesos.

Una cola (queue) permita la comunicación de dos o más procesos. Esta es una abstracción fundada en las tuberías descritas anteriormente. Una fila de datos apilados por distintos procesos en un buffer va siendo retirada por otros de manera asíncrona. La Fig. 2 ilustra el flujo de datos presenta en la comunicación entre procesos usando filas.

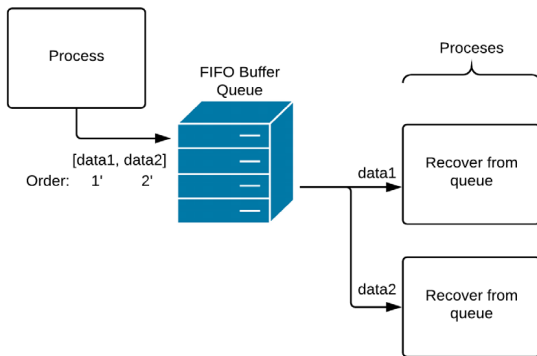


Figura 2. Diagrama de flujo de datos a través de una fila (queue) entre procesos.

2.3. Arquitectura de software

Entre los métodos enunciados en la subsección anterior, la forma de comunicación entre procesos que más nos conviene es mediante sockets.

2.3.1. IPC: Unidades de lectura y procesamiento

Recordemos la topología **Publisher – Subscriber** de la subsección anterior referente a ZMQ. Si podríamos generalizar este patrón para interconectar múltiples publicadores con múltiples suscriptores al mismo tiempo y que los mensajes lleguen al destino definido (subscriber) solo por medio del filtrado de tópicos (inherente a cada publicador), tendremos la posibilidad de instanciar unidades de lectura que publiquen de manera asíncrona y paralela datos leídos a unidades de procesamiento específicas. Cada unidad de lectura tiene un

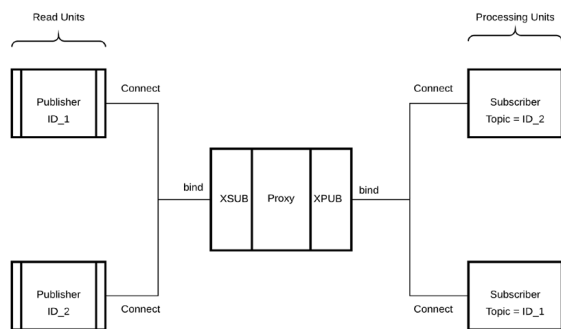


Figura 3. Topología de múltiples publicadores y múltiples suscriptores interconectados [4].

identificador (ID) único que será el mismo que su tópic de publicación. Las unidades de procesamiento que quieren recibir lo que una unidad de lectura publique solo deben configurar en su socket el tópic de escucha igual al tópic de publicación descrito previamente.

Esto solo se consigue si agregamos un proxy. Este está conformado por dos sockets tipo XSUB y XPUB que se vinculan a dos direcciones (ipc o tcp) de manera invariable. Son los publicadores y suscriptores los que se conectan a los sockets XSUB y XPUB de manera correspondiente y estos se encargan de transmitir los mensajes. La Fig. 3 ilustra lo previamente descrito. Es importante recalcar que el proceso en donde se ejecuta el proxy es el del controlador.

2.3.2. IPC: Unidades de procesamiento

Luego de haber recibido la data proveniente de una unidad de lectura, una unidad de procesamiento puede tener otras unidades más esperando la data modificada de la misma. Se mantuvo el patrón **Publisher – Subscriber** para comunicación entre unidades de procesamiento.

2.3.3. IPC: Unidades y operaciones

Nótese que una unidad de lectura es una unidad de procesamiento y también puede tener operaciones a ejecutar antes de publicar data. La configuración usada en este escenario no ha sido descrita anteriormente y se denomina **Dealer – Router** [4]. Por definición, una operación recibe un objeto (data) y ejecuta una serie de métodos que pueden ser de manera instantánea o progresiva (acumulativa). En ambos casos, siempre retorna un objeto (data) modificado (operación instantánea) o sin alterar (operación acumulativa) hasta que se cumpla una condición. Las unidades de procesamiento poseen un diccionario con todas las operaciones que deben ejecutar de manera ordenada.

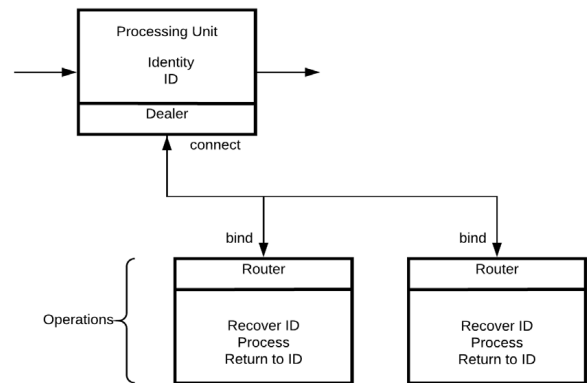


Figura 4. Topología referente a la comunicación entre unidades y operaciones.

Además de las dos interfaces de comunicación que posee una unidad de procesamiento (PUB y SUB), también posee otra denominada DEALER. Esta posee la característica de tener una identidad. Así, cuando esta envía un mensaje, el que la recibe sabe quién se lo envió. Los procesos de operación tienen en su interfaz de comunicación un socket de tipo ROUTER. Con este socket identifica la identidad de los sockets DEALER que le transmitieron un mensaje y por ende sabe de qué unidad se trata o quien quiere conversar con él. La Fig. 4 ilustra lo descrito previamente.

2.3.4. Escalabilidad

La integración de las topologías de comunicación resulta en una arquitectura de software modular definida capaz de ser desplegada en distintas plataformas. Gracias a ello, tienen la propiedad de ser escalable. Experimentos sencillos podrán ser

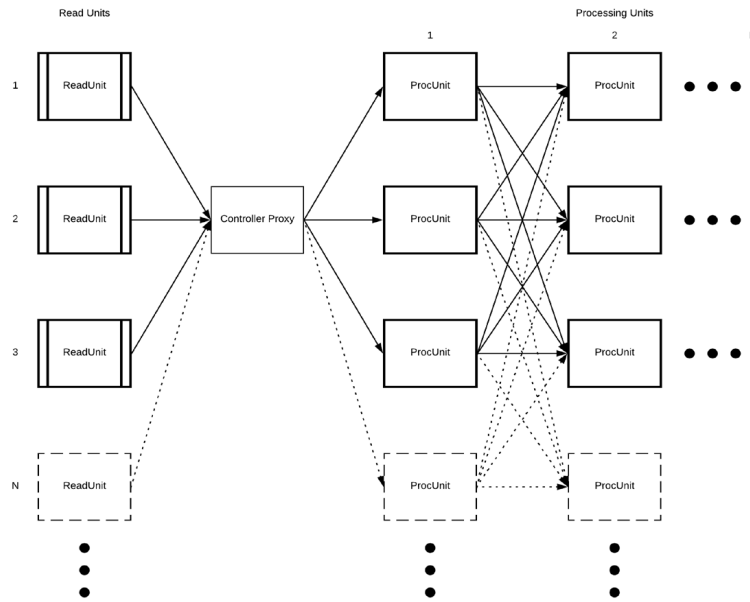


Figura 5. Arquitectura de software paralela y escalable. No se incluye en este diagrama a las operaciones

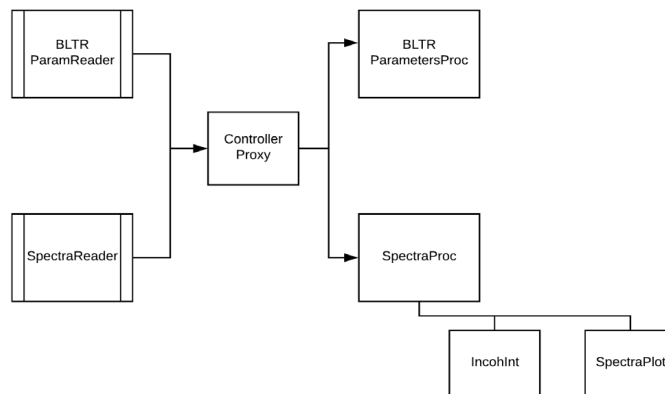


Figura 6. Arquitectura de un experimento de prueba

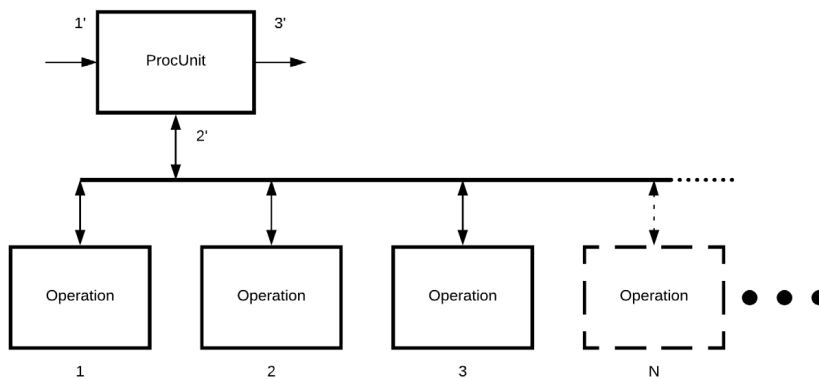


Figura 7. Arquitectura de software relevante a las unidades y operaciones

ejecutados en una sola unidad de cómputo, y los que requieran mayor capacidad de computación podrán ser descritos en múltiples plataformas (*cluster*) derivando en una concurrencia aún mayor. Las figuras 5 y 7 muestran como la arquitectura puede desarrollarse en relación con las características del experimento en cuestión.

3. RESULTADOS

Las pruebas se hicieron en una PC del Radio Observatorio de Jicamarca. Presentaba un procesador Intel® Core™ i5-650 (2 núcleos), 4 GB de RAM y OS Ubuntu 16.04. Los programas están descritos en Python 2.7.14. Uno de los experimentos se ilustra en la Fig. 6.

4. DISCUSIÓN

Las operaciones presentes en la herramienta Signal Chain, se pueden diferenciar en 2 tipos: externas e internas. Las operaciones internas esta definidas dentro de la clase de una unidad de procesamiento y solo está la puede invocar y utilizar. Las operaciones externas se encuentran definidas aparte y cualquier unidad puede acceder a utilizarlas. Dado que algunas operaciones internas no requieren de una capacidad de cómputo significativa, no son elevadas a la categoría de proceso y se les maneja de manera interna por el decorador. Muchas veces en un experimento, una misma operación forma parte de varias unidades de procesamiento y se instancia más de una vez (arquitectura anterior). Como trabajo a futuro se tiene como objetivo modificar la configuración de las operaciones para que estas no solo trabajen con una unidad, sino con varias evitando así una múltiple instanciación de esta.

Es conveniente analizar un caso relevante a la interacción entre las unidades de procesamiento y las operaciones. Tal como está descrita la nueva arquitectura, la unidad de procesamiento debe esperar a que la operación termine y regrese el objeto que se le transmitió (ya sea el mismo o no). Por lo que si la operación tarda en ejecutarse de manera significativa atrasara la ejecución de las demás operaciones y demás unidades de procesamiento que dependen de ello, formándose un cuello de botella en el flujo de datos.

5. CONCLUSIONES

Una nueva arquitectura de software ha sido presentada para la herramienta de procesamiento de señales Signal Chain.

La forma más eficiente de intercambiar información entre procesos es mediante Sockets. ZMQ presenta soluciones ingeniosas para cada posible escenario de trabajo.

La nueva arquitectura es compatible con lo ya descrito en Signal Chain. La mayoría de cambio se dio en los archivos base mas no las unidades y operaciones en sí. Por lo que no hubo que reestructurar todo lo ya establecido por la herramienta.

6. RECOMENDACIONES

Las pruebas de rendimiento deben ser ejecutadas en una plataforma/hardware que cumpla con condiciones mínimas relevantes al número de procesos a instanciar para que realmente exista un paralelismo de procesos.

7. AGRADECIMIENTOS

Agradezco al Instituto Geofísico del Perú a través del Radio Observatorio de Jicamarca y a Ciencia Internacional por brindar la oportunidad a estudiantes de pregrado de participar en proyecto de ingeniería y comprometerse en la formación de ingenieros de calidad en el Perú.

8. BIBLIOGRAFÍA

- [1] Signal Chain Wiki: Developer's manual. Recuperado de: http://jro-dev.igp.gob.pe:3000/projects/signal-chain/wiki/Manual_de_Desarrollador#Developers-manual
- [2] Hintjens, P. (2013). ZeroMQ: messaging for many applications. " O'Reilly Media, Inc."
- [3] Kurose, J. F., & Ross, K. W. (2009). Computer networking: a top-down approach (Vol. 4). Boston, USA: Addison Wesley.
- [4] Hintjens, P. ØMQ - The Guide. Obtenido de: <http://zguide.zeromq.org/page:all>

DISEÑO DE UNA INTERFAZ DE CONTROL Y SUPERVISIÓN PARA TRANSMISORES ATRAD

**Design of a control and supervision interface for ATRAD transmitters*

George Steve Fajardo Soria

Universidad Nacional de Ingeniería
Facultad de Ingeniería Mecánica
george.fajardo@jro.igp.gob.pe

Asesor: Ing. Rommel Yaya

Radio Observatorio de Jicamarca - IGP

RESUMEN

El Instituto Geofísico del Perú, a través de su Radio Observatorio de Jicamarca, realiza estudios del espacio cercano mediante el uso de técnicas de radar, para ello cuenta con un arreglo de antenas de 18,432 dipolos, un sistema de adquisición de alta sensibilidad y amplificadores de alta potencia como sus principales herramientas. El presente proyecto tiene como objetivo el diseño de una interfaz de control y supervisión para los amplificadores de estado sólido ATRAD que forman parte del sistema de amplificadores de alta potencia del ROJ; para ello disponemos de una librería que implementa el protocolo ARCP (Atrad Radar Control Protocol) desarrollada por el fabricante de estos equipos y que servirá de base para el desarrollo del presente trabajo. En proyecto tiene como elemento principal al Raspberry pi 3 el cual servirá de host y donde se implementarán los programas de la interfaz.

Palabras clave: ATRAD, transmisor de estado sólido, sistema de control y monitoreo.

ABSTRACT

The Geophysical Institute of Peru, through its Jicamarca Radio Observatory (JRO), study the space using radar techniques. This radar system consists of an antenna array of 18,432 dipoles, a high sensitivity acquisition system, and high-power amplifiers as its main components. The objective of this project is to design a control and supervision interface for the ATRAD solid state amplifiers that are part of the JRO's high power amplifier system; for this we have a library that implements the ARCP protocol (Atrad Radar Control Protocol) developed by the manufacturer and is taken as a starting point for the development of this work. The main feature of the project is the Raspberry pi 3 which will be the host and where the interface programs will be implemented.

Keywords: ATRAD, solid state transmitter, control and monitoring system.

Citar como: Fajardo, G. y Yaya, R. (2019). Diseño de una interfaz de control y supervisión para transmisores ATRAD. Compendio de estudiantes 2019, IGP.

1. AMPLIFICADORES DE ESTADO SÓLIDO

No todos los amplificadores son iguales y existe una clara distinción entre la forma en que se configuran y operan las etapas de salida. Las principales características operativas de un amplificador ideal son la linealidad, la ganancia de señal, la eficiencia y la potencia de salida, pero en los amplificadores del mundo real siempre hay una compensación entre estas diferentes características, actualmente existen 2 tipos de amplificadores, los amplificadores de válvulas o de tubos de vacío y los amplificadores de estado sólido.

Las principales diferencias son que los amplificadores de válvulas soportan mucha mayor potencia (orden de los MW), sin embargo, la desventaja es que la onda no puede ser interrumpida drásticamente, ello se traduce en una codificación pobre.

Los amplificadores sólidos consisten básicamente en un arreglo de transistores, la señal es dividida aumentada en potencia y luego es unida de nuevo.

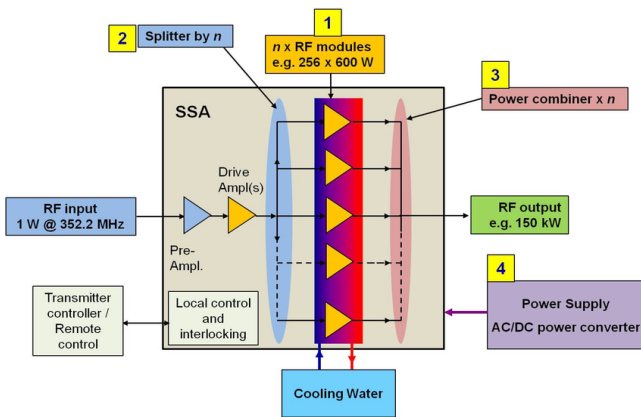


Figura 1: Esquema eléctrico de un amplificador de estado sólido.

2. REQUERIMIENTOS DEL SISTEMA

El sistema consiste de un transmisor de estado sólido ATRAD, el transmisor puede operar durante los experimentos muchas horas ininterrumpidas en las cuales debe ser vigilado, sin embargo, en ciertos turnos no hay personal suficiente como para vigilar exclusivamente el estado del transmisor por ello el sistema debe poder ser monitoreado y controlado remotamente.

Los datos además deben ser guardados para su posterior análisis y también deben ser mostrados en tiempo real en una interfaz amigable.

3. DISEÑO E IMPLEMENTACIÓN

3.1. Software y Hardware utilizados

La solución que se detallará posteriormente, utiliza los siguientes recursos:

- Transmisor ATRAD
- RASPBERRY
- RASPBIAN
- LIBRERÍA ARCP
- Servidor LAMP(Linux-Apache-MySQL-PHP)
- Python
- C++

3.2. Solución planteada

Para el presente proyecto se planteó el uso de una Raspberry Pi que servirá para monitorear constantemente y controlar cuando se requiera al ATRAD.

Para explicar de manera más sencilla se puede entender el desarrollo como dos partes, la lectura y la escritura, teniendo en cuenta que se trabajará con un programa en C++ en la raspberry, un programa en python en la raspberry, una base de datos Mysql en la raspberry, un servidor local (host) en Apache en la raspberry, un archivo CSV y un archivo TXT en el servidor local.

La raspberry mediante el uso de la librería ARCP, adquiere información de sus sensores, guardando las variables leídas en un archivo CSV mediante el programa en C++, luego el programa en python en la raspberry toma los datos desde el archivo CSV y los escribe en la base de datos, al final la raspberry también sirve de host, es así que una página web lee la base de datos y muestra los datos.

Para el control del transmisor ATRAD, el programa de python también sensa constantemente el estado de la habilitación(enable) en la página web, que sirve para iniciar una secuencia de pulsos, escribe este dato "enable" en el archivo TXT y el programa de C++ toma el dato desde el archivo TXT para luego ser enviado al ATRAD.

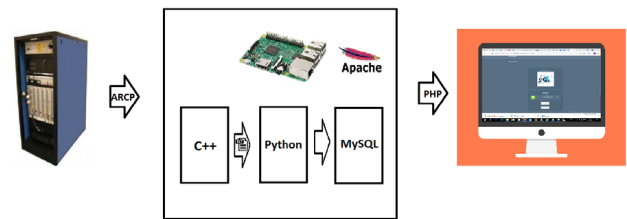


Figura 2: Esquema general de la solución.

3.3. Implementación

El desarrollo consistió en interconectar los scripts anteriormente mencionados.

El primer programa que se desarrolló fue el programa en C++, este programa parte desde el punto que tenemos la conexión ya realizada, si la conexión ya está realizada se tendría que acceder a sus variables y mostrarlas, esto se logró revisando la estructura de cada variable, por ejemplo para acceder a la "potencia loss" de uno de los slots, se tiene que acceder primero a la estructura "sysstat", esta estructura contiene 2 variables y una estructura, esta estructura es denominada union data, dentro de union data encontramos otras dos estructuras denominadas STX2 y BSM, dentro de la estructura STX2 es donde se encuentra la variable temperatura, que finalmente es la que muestra nuestro programa.

Para detallar un poco más el programa se mencionará los métodos usados.

Para poder escribir dentro de un archivo csv se tiene la función #include <fstream> que es la que en este caso permite el acceso a este tipo de archivo.

```
ofstream archivotexto;
    archivotexto.open
    ("ATRADvalues.csv");
```

El método para realizarlo es primero abrir el archivo luego se tiene acceso a él, escribiendo los datos en la base de datos, se escribe en el archivo mediante el uso de

```
archivotexto<<sysstat->data.stx2-
>rf_card_stat[i].output_stat-
```

```
>forward_power<<"";
```

De esa forma el programa guarda los datos en el archivo csv.

El programa también debe revisar el estado de enable, eso lo hace leyendo el archivo de texto, luego es actualizado en el ATRAD utilizando el valor obtenido y el handle.

Luego tenemos los programas en python, el primer programa usa el método GET para obtener el valor del enable desde el enlace que genera el valor del dato y luego el segundo programa que toma los datos del archivo CSV y los sube a la base de datos.

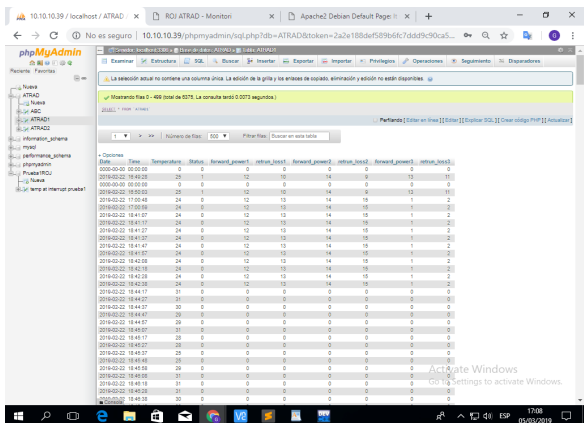


Figura 3: Base de datos en MariaDb vista desde phpmyadmin (datos reales).

Finalmente tenemos la interfaz web que cuenta con 2 botones, enable y disable, los cuales habilitan y deshabilitan el ATRAD para la transmisión de pulsos, de forma predeterminada "enable" habilita la transmisión mediante fuente externa, además la web tiene la opción de redireccionar hacia un gráfico valor-tiempo del dato deseado y también hacia la base de datos mediante phpmyadmin, finalmente la interfaz web muestra una tabla con los principales valores monitoreados.

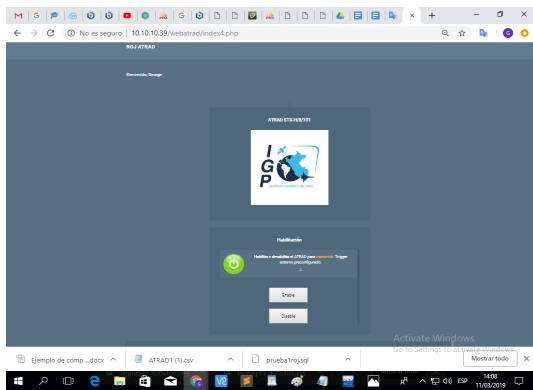


Figura 4: Vista superior de la interfaz web.

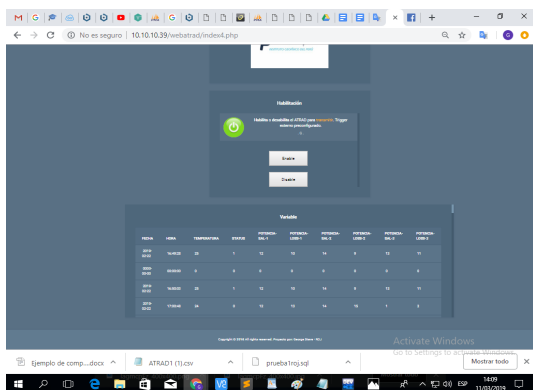


Figura 5: Vista inferior de la interfaz web.

4. RESULTADOS

Se realizaron diversas pruebas del sistema de recarga, con la finalidad de conocer el comportamiento de cada una de las etapas y realizar observaciones para futuras mejoras.

22/02/2019	19:01:17	31
23/02/2019	19:01:20	30
24/02/2019	19:01:23	27
25/02/2019	19:01:26	29
26/02/2019	19:01:29	28
27/02/2019	19:01:32	28
28/02/2019	19:01:35	26
01/03/2019	19:01:38	27
02/03/2019	19:01:41	30
03/03/2019	19:01:44	27
04/03/2019	19:01:47	31
05/03/2019	19:01:50	28
06/03/2019	19:01:53	26
07/03/2019	19:01:56	27
08/03/2019	19:01:59	26
09/03/2019	19:02:02	29

Tabla 1. Valores experimentales

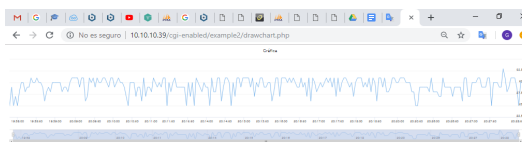


Figura 6: Gráfico Temperatura vs Tiempo.

5. CONCLUSIONES Y RECOMENDACIONES

Los programas realizados luego de ser probados durante periodos de 5 horas funcionaron correctamente, registrando los datos de Fecha, Hora, Temperatura , Status, Potencia Emitida y Potencia perdida de cada módulo cada 3 segundos aproximadamente.

La interfaz web muestra una tabla donde se puede ver todos los datos ordenados históricamente, desde el primero hasta el último.

La tarjeta de desarrollo puede funcionar tiempos prolongados consumiendo poca energía, además debido a su tamaño reducido, es fácil de instalar en cualquier lugar.

En general el desarrollo realizado es un prototipo funcional que ya puede ser usado en los transmisores ATRAD que se tienen operando actualmente, permitiendo así un control y monitoreo remoto.

Conectar este desarrollo al SIR (Sistema Integrado de Radar) para que se pueda acceder a la web desde todo el ROJ.

Desarrollar una web con más dinámica, posiblemente con JavaScript que permita recargar automáticamente los datos cada cierto tiempo sin afectar la visibilidad de la web, además de incorporar gráficos.

6. AGRADECIMIENTOS

Deseo agradecer al Instituto Geofísico el Perú, a través del Radio Observatorio de Jicamarca por facilitar los recursos necesarios para la realización del presente desarrollo, el cual sirvió para completar mis prácticas pre-profesionales.

A su vez, agradecer al Ing. Rommel Yaya por su apoyo y motivación constante.

7. BIBLIOGRAFÍA

[1] Jacob, J. (2016). Radio Frequency Solid State Amplifiers. 10.5170/CERN-2015-003.197.

[2] Aradhana Kumar and Vaibhav Mathur (2018). A beginner's guide to getting started with the IoT by using a Raspberry Pi to connect devices. Recuperado de: <https://maker.pro/raspberry-pi/tutorial/how-to-get-started-with-iot-using-raspberry-pi-and-putty-part-1>

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE RADAR IONOSONDA USANDO UN HARDWARE DE CÓDIGO ABIERTO

**Design and implementation of an ionosonde radar system using an open-source hardware*

Francisco Sulca Jota

Universidad Nacional de Ingeniería
Facultad de Ingeniería Eléctrica y Electrónica
fsulcaj@uni.pe

Asesor: Ing. Joaquín Verástegui

Área de IDI - Radio Observatorio de Jicamarca - IGP

RESUMEN

En el presente trabajo se hace una pequeña descripción del diseño de un sistema de radar para una Ionosonda en un dispositivo de código abierto. El sistema abarca una larga cantidad de temas para su construcción; entre los que se encuentran los tópicos de sistemas de radar, sistemas embebidos, tecnología System on Chip, FPGAs, ASICs, Linux, lenguajes de programación: C y Python, lenguajes de descripción: VHDL y Verilog, lenguaje de comandos: tcl, entornos de desarrollo de hardware: Vivado HLx (en sus diferentes versiones) entornos de desarrollo de Hardware: Xilinx SDK y Vitis, entorno de desarrollo de sistemas embebidos: Petalinux, entre otros. El sistema de radar Ionosonda se implementa utilizando la tecnología System on Chip para su creación. Consiste en crear un sistema de radar sencillo que consiste en un transmisor, un receptor y un sistema de almacenamiento de datos. El control de la arquitectura es desde un sistema operativo Linux que se ejecuta en el mismo dispositivo, en conjunto con una memoria micro SD externa.

Palabras clave: [System on Chip (SoC)]

ABSTRACT

This work presents the design of an Ionosonde radar system based on open-source hardware. The system involves a large number of topics for its construction like radar systems, embedded systems, System on Chip technology, FPGAs, ASICs, Linux, programming languages: C and python, description languages: VHDL and Verilog, command language: tcl, development environments Hardware: Vivado HLx (in its different versions) Hardware development environments: Xilinx SDK and Vitis, embedded system development environment: Petalinux, among others. The Ionosonde radar system is implemented using System on Chip technology and consists of a transmitter, a receiver, and a data storage system. The framework control is a Linux based operating system that runs on the same device with the aid of an external micro SD memory.

Keywords: [System on Chip (SoC)]

1. INTRODUCCIÓN

En la actualidad, año 2019, el desarrollo tecnológico de los dispositivos System on Chip (SoC) están sobresaliendo sobre otras tecnologías, de hecho, cada vez son más los centros de investigación que utilizan esta tecnología. Se puede resumir estos dispositivos en que fusionan las ventajas de construcción de hardware de un FPGA con las de un procesador en un mismo chip.

En el Radio Observatorio de Jicamarca se está utilizando esta tecnología para crear un sistema de radar lonosonda que se implementará en una tarjeta Zynq "Red Pitaya" con la que se cuenta en la sede.



Figura 1. Chip de un dispositivo Zynq 7000 System on Chip de Xilinx.

1.1 VITIS Platform

Creada por Xilinx, es una plataforma de desarrollo de software para sistemas embebidos. Permite desarrollar programas en los lenguajes C, C++ o Python. Es la última plataforma que reemplaza a las ya sin actualización Xilinx SDK, SDSoC and SDAccel Development Environment.

1.2 Vivado Design Suite

Creado por Xilinx, es una plataforma de desarrollo de hardware, mediante su interfaz gráfica o de consola, por medio del uso de comandos TCL, se puede crear arquitecturas para dispositivos FPGAs o SoCs. Permite el uso combinado de los lenguajes de descripción VHDL y Verilog, además que facilita el uso de IP cores para acelerar el proceso de diseño.

1.3 IP cores

Bloques escritos en lenguaje HDL (VHDL o Verilog) que realizan una o varias funciones, son útiles porque se pueden reutilizar y ahorran tiempo en el diseño de arquitecturas.

1.4 Zynq-board

Son tarjetas que llevan un dispositivo SoC Zynq de Xilinx en ellas. Son diversas y pueden ser creadas para propósitos generales y de entrenamiento como las tarjetas Zedboard o Zybo-Z7, o para propósitos específicos, por ejemplo, procesamiento de datos, imágenes o aplicaciones en RF como la Red Pitaya.

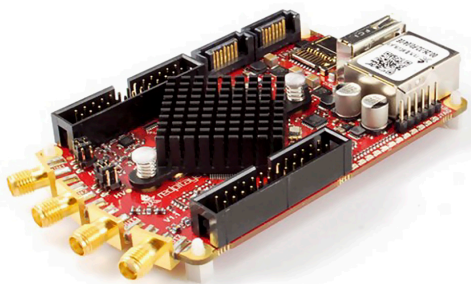


Figura 2. La imagen superior muestra la tarjeta de desarrollo Red-Pitaya de STEAMLab en donde se implementa el sistema de radar lonosonda.

2. DESARROLLO

2.1. Parte de Hardware: Construcción de la arquitectura del Sistema de Radar

El Sistema de Radar, en la parte de hardware, está conformado por lo siguiente:

- **Transmisor:** Utilizando un convertor digital a analógico (DAC) de la tarjeta Red Pitaya se construye un generador de señales en el FPGA que dará forma al mensaje que se desea enviar. -Una particularidad del radar lonosonda es que realiza un barrido de frecuencias de entre 0.5 MHz a 30 MHz-.
- **Receptor:** por medio de un convertor analógico a digital (ADC) de la tarjeta Red Pitaya se realiza la recepción. Se creará una arquitectura, en el FPGA, para los procesos de filtrado y demodulación de la señal recibida.
- **Almacenamiento de Data:** Envío de la cadena de bits de la señal demodulada a la memoria DRAM del dispositivo Zynq.

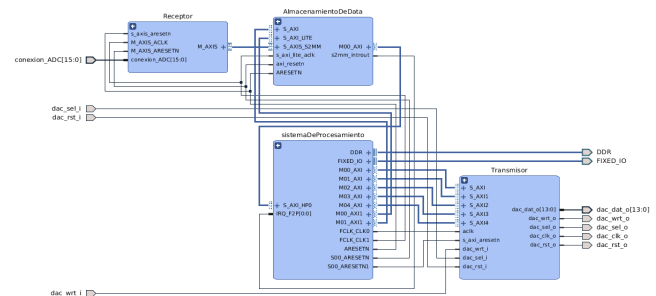


Figura 2. La imagen superior muestra la tarjeta de desarrollo Red-Pitaya de STEAMLab en donde se implementa el sistema de radar lonosonda.

2.2. Parte de Software: Ingreso de parámetros de configuración de Hardware

En la parte de software el sistema de radar contempla lo siguiente:

- **Escritura en la memoria BRAM de la parte de hardware del transmisor.** Con lo cual se configura el funcionamiento de la arquitectura del transmisor. -Se escriben todos los valores que van a intervenir en el proceso de barrido de frecuencias que realiza la lonosonda-.
- **Envío de los parámetros de configuración que dan forma a la señal que se desea enviar,** como el IPP, variación del barrido de frecuencias, etc.
- **Creación del programa que traslade los requerimientos de funcionamiento del sistema de radar lonosonda a un conjunto de cadenas de bits y de registros que reconfiguran al hardware.** -Se configura un radar lonosonda con parámetros de longitud de ondas, frecuencia, etc. Se tiene que cambiar estos parámetros, por medio de ecuaciones físicas, a número de ciclos de reloj que es como se trabaja a nivel de hardware-.

2.3. Parte de Sistema: Sincronía entre las partes de Hardware y Software

Las partes de PS (Processing System) y PL (programmable logic) en un dispositivo Zynq estas estrechamente relacionadas, dado que estos dispositivos son diseñados para que estas dos partes puedan trabajar en forma conjunta de una manera óptima. Sin embargo, aún se presentan fallas en estos sistemas embebidos que hacen que la comunicación PS y PL tenga errores y se presente ruido no deseado.

Para el diseño del sistema de radar se tienen presentes estas características así que se toma la decisión de minimizar los puntos débiles de esta tecnología -cabe mencionar que en el mercado ya existen dispositivos especializados, de mayor calidad y de mucho mayor costo que solucionan en gran parte este problema (llamados RFSoc)-

Los criterios de diseño que se utilizan son los siguientes:

- Transmisión: Escritura de los parámetros de configuración en la memoria BRAM de la parte de FPGA del Zynq.
- Cálculo de los parámetros de configuración en el procesador (PS) -resolución de fórmulas físicas y cambio de base a las soluciones para que pueden ser escritas y entendidas por la parte de Hardware (PL)-
- Recepción: La configuración que se utiliza es la de DMA en modo de operación de Scatter-Gatter. Es una configuración bastante compleja por lo que en resumen se puede decir que optimiza la comunicación entre PS y PL. dado que todo el trabajo del envío de data proveniente del demodulador es enviado directamente a ser almacenada en la DRAM sin que intervenga en el procedimiento, en lo mayor posible, el procesador (PS) y de esta forma evitar el error que se produce cuando se trabaja en conjunto la parte de procesamiento con la parte lógica.

3. RESULTADOS

El hardware que se tienen que agregar parte por parte a la arquitectura del sistema de radar han sido estudiado y probado en tarjetas Zynq de desarrollo, que son tarjetas que permiten el aprendizaje de los diferentes componentes. Se usan estas tarjetas y no se trabaja directamente con la tarjeta de implementación Red Pitaya dado que brindan mayor soporte en cuando errores comunes, mayor material de estudio como proyectos específicos para aprender el funcionamiento de determinados componentes, etc.

3.1 Sistema de Transmisión

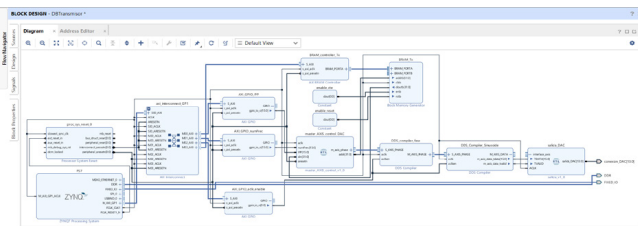


Figura 4. Diagrama de bloques del sistema de transmisión. Cada bloque cumple una función en la generación de la señal, destaca la configuración de los DDS Compilers que forman el mensaje, para su funcionamiento necesitan de la escritura en los registros.

BLOCK DESIGN - DBTransisor *

Cell	Slave Interface	Slave Segment	Offset Address	Range	High Address
PS7					
Data (32 address bits : 0x80000000 [1G])					
AXI_GPIO_IPP	S_AXI	Reg	0x8120_0000	64K	0x8120_FFFF
AXI_GPIO_numfrec	S_AXI	Reg	0x8121_0000	64K	0x8121_FFFF
AXI_GPIO_aclik_enable	S_AXI	Reg	0x8122_0000	64K	0x8122_FFFF
BRAM_controller_Tx	S_AXI	Mem0	0x8000_0000	8K	0x8000_1FFF

Figura 5. Lista de las direcciones de registros y memoria que configuran el transmisor.

3.2 Sistema de Recepción y Almacenamiento

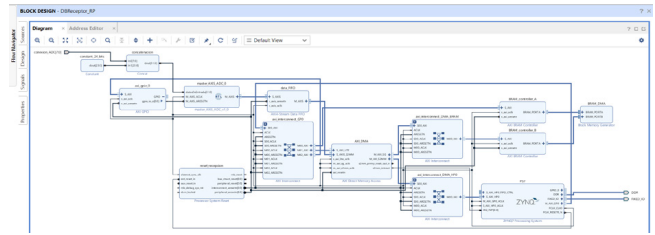


Figura 6. Diagrama de bloques del sistema de recepción. El bloque que corresponde al AXI DMA permite enviar los datos recibidos a la memoria BRAM.

BLOCK DESIGN - DBReceptor_RP

Cell	Slave Interface	Slave Segment	Offset Address	Range	High Address
AXI_DMA					
Data_S2MM (32 address bits : 4G)					
PS7	S_AXI_HP0	HP0_DDR_LOWOCM	0x0000_0000	512M	0x1FFF_FFFF
Data_SG (32 address bits : 4G)					
BRAM_controller_B	S_AXI	Mem0	0x4000_0000	8K	0x4000_1FFF
PS7					
Data (32 address bits : 0x40000000 [1G])					
AXI_DMA	S_AXI_LITE	Reg	0x4040_0000	64K	0x4040_FFFF
axi_gpio_0	S_AXI	Reg	0x4120_0000	64K	0x4120_FFFF
BRAM_controller_A	S_AXI	Mem0	0x4000_0000	8K	0x4000_1FFF

Figura 7. Lista de memoria y registros que configuran al sistema de recepción.

La imagen superior los IP cores resaltantes son: AXI DMA para el envío de data a la memoria DRAM.

3.3 Configuración de memoria y registros

La imagen superior muestra las direcciones de memoria y registros que se implementan en la parte de hardware. Con la manipulación de estos registros se puede cambiar el funcionamiento del hardware.

3.4 Sistema de Radar Completo

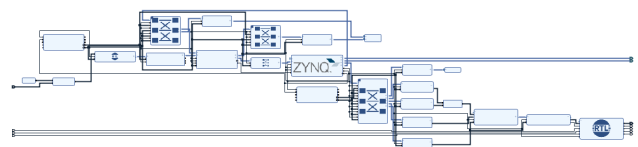


Figura 8. Cuando se trabaja en el diseño de hardware la cantidad de bloques que se utilizan puede llegar a ser enorme, mayor a 100 cores. Lo cual implica una mayor cantidad de recursos cada vez que crece los circuitos que se están colocando.

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DIGITAL PARA LA MEDICIÓN DEL DESFASE ENTRE DOS SEÑALES

**Design and implementation of a digital system for the measure of the difference of phase between two signals*

Isaac Mario Tupac Dávila

Universidad Nacional Mayor de San Marcos
Facultad de Ingeniería Electrónica , Eléctrica y
Telecomunicaciones
isacctd92@gmail.com

Asesor: Ing. Joaquín Verástegui

Área de investigación , desarrollo e innovación
I&D+I
Radio Observatorio de Jicamarca - IGP

RESUMEN

Este nuevo diseño será utilizado para medir el desfase entre dos señales de la misma frecuencia, la señal producida en el generador y la señal que llega a la antena principal del Radio Observatorio de Jicamarca. La frecuencia a la que trabaja la antena principal es de 50 MHz y dependiendo del desfase se producirá el patrón de radiación requerido en la dirección requerida. Actualmente, el Radio Observatorio de Jicamarca en el campo detecta desfases mínimos de 10 grados. Lo que hará este diseño es mejorar la detección del desfase y detectar desfases de hasta 1 grado de precisión para así ser más eficaces en la práctica. Este diseño es solo una etapa ya que además se necesitarán Directional Coupler y atenuadores para bajar la potencia de la señal que recibirá, los cuales se desarrollarán en otros proyectos del Radio Observatorio.

Palabras clave: Desfase, patrón de radiación, grados

ABSTRACT

This new design will be used to measure the phase difference between two signals of the same frequency, the signal produced in the generator and the amplified signal that is transmitted through the main antenna of the Jicamarca Radio Observatory. The operation frequency of the radar is 50 MHz and depending of the phase difference will produce the desired radiation pattern in the desired direction. Nowadays, the minimum phase difference detected in the field is 10 degrees. This design will improve the phase detection with a precision up to 1 degree. This design is only one stage, because for a correct implementation, it will also need a directional coupler and attenuators to reduce the power of the transmitted signal.

Keywords: Phase difference, radiation pattern, degrees

1. INTRODUCCIÓN

El Radio Observatorio de Jicamarca (ROJ) alberga como radar principal una antena del tamaño de 300 x 300 m², conformada por un arreglo de 18 432 dipolos lineales. Para controlar la longitud del cable que se encuentra entre el generador y la antena de una manera más rápida, se implementó el ABS (Automatic Beam Steering). Dependiendo del desfase entre la señal que se produce en el generador (señal que entra al ABS) y la que recibe la antena (señal que sale del ABS) se produce un patrón de radiación con una determinada dirección.

Actualmente se quiere mejorar la detección del desfase entre esas dos señales con un nuevo diseño electrónico. En este informe se explicará el fundamento teórico en el cual se basa este diseño, las simulaciones desarrolladas en el simulador LTSpice XVII, la gráfica de respuesta obtenida por la simulación, el diseño de implementación elaborado en Eagle 9.3.1 y las pruebas en el osciloscopio.

2. DESARROLLO

2.1. Diagrama de Bloques proyectado

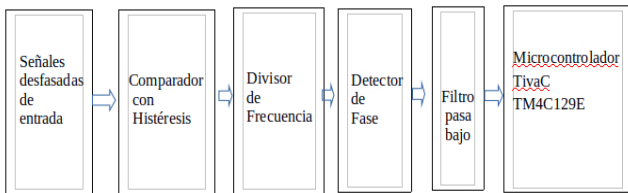


Figura 1. Diagrama de Bloques

2.2. Detector de Fase

Para detectar el desfase se utilizó el detector de fase de tres estados. Este diseño está implementado por dos Flip Flops tipo D y una compuerta AND como se muestra a continuación

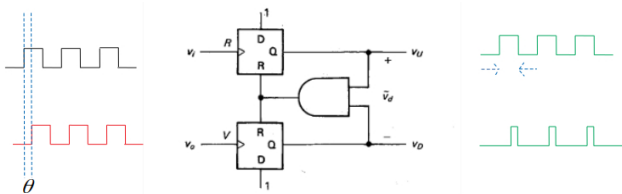


Figura 2: Detector de Fase de tres estados

¿Cómo funciona? Dependiendo del desfase de las señales que entran por los pines del clock de los flip-flops. Por ejemplo, si la señal V_o está desfasada con respecto a la señal V_i un ángulo θ, se generará una señal cuadrada en el pin de salida VU como se ve en la Figura 2. En la salida VD se generarán pulsos, como se muestra en la Figura 2.

¿Cómo se relaciona el desfase con los voltajes de salida?

Para encontrar la relación tenemos que definir V_d.

V_d es la resta de las señales continuas generadas por las ondas cuadradas y los pulsos correspondientes a VU y VD, así que V_d = VU - VD. Estas señales continuas se generarán ya que VU y VD pasarán cada una por un filtro pasa bajo.

Entonces si el desfase es pequeño, el ancho de las ondas cuadradas serían pequeñas produciendo a la salida de los filtros una señal en continua pequeña, ya que la ecuación que describe el valor de la señal continua depende del área de la señal. Entonces el valor de V_d sería pequeño.

Si el desfase es grande, el ancho de las ondas cuadradas sería grande, produciendo a la salida de los filtros una señal con valor de voltaje continuo grande. Entonces el valor de V_d sería grande

Si nos damos cuenta, el voltaje V_d es directamente proporcional al desfase, con lo cual su relación es teóricamente lineal. Gráficamente lo podemos definir y expresar como:

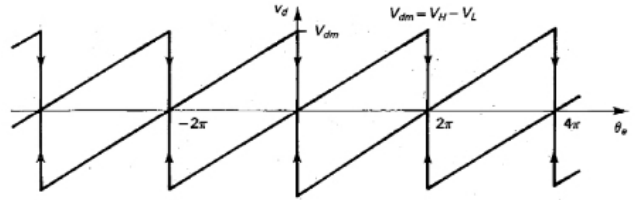


Figura 3: Respuesta gráfica de V_d con θe respecto a

$$V_d = K_d \theta_e + V_{d0} \quad \text{Donde:} \quad K_d = (V_H - V_L) / 2\pi$$

En la práctica V_H en nuestro caso es 3.3v y V_L es 0v. V_{d0} es el voltaje inicial, cuando las dos señales tienen un desfase de 0 grados. El voltaje V_{d0} tiende a ser siempre 0v.

Como vemos el comportamiento de la gráfica es positiva y negativa. Esto se debe a que la señal cuadrada puede originarse por V_U o V_D. Si se originara por V_U, la resta para hallar V_d (V_d = V_U - V_D) sería positiva, pero si se originara por V_D, la resta para hallar V_d sería negativa.

2.2.1. Respuesta gráfica de V_d con respecto a θe a altas frecuencias

Ya que se está tomando en cuenta los tiempos de delays a altas frecuencias, el tiempo de respuesta será un poco más retrasada con respecto a lo ideal.

- Observación

Para más detalles acerca de la base teórica del funcionamiento, revisar el informe completo.

3. SIMULACIÓN EN LTSPICE

3.1. Diseño en LTSpice

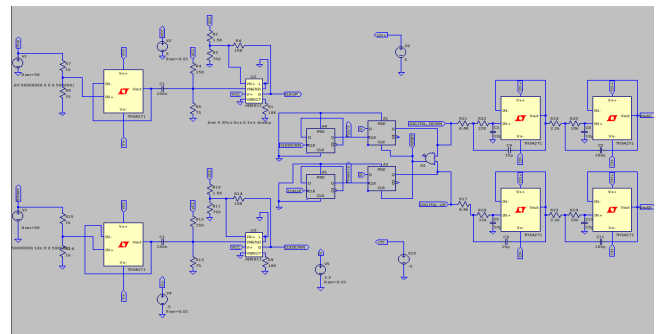


Figura 4: Diseño en LTSpice

Este diseño está compuesto por 3 etapas:

La primera etapa es un comparador por Histéresis. Es el circuito que se encuentra dentro del cuadrado rojo en la figura 4. Se utiliza uno para cada señal y su función es convertir la señal sinusoidal de entrada en ondas cuadradas.

La segunda etapa es el divisor de frecuencia y el detector de fase. Es el circuito que se encuentra en el cuadrado negro de la figura

4. En esta etapa el divisor de frecuencia genera señales con una frecuencia de entrada dividida entre dos, manteniendo el mismo desfase. El detector de fase recibe estas señales y genera en sus respectivas salidas las ondas cuadradas y ondas pulsantes como se describió en el inciso 2.2.

La tercera etapa es la etapa de filtro. Es el circuito que se encuentra en el cuadrado celeste de la figura 4. En esta etapa se usan dos arreglos de filtros Butterworth de cuarto orden cada uno para hallar las señales en continua.

4. DISEÑO EN EAGLE

4.1 Etapa de comparación con Histéresis

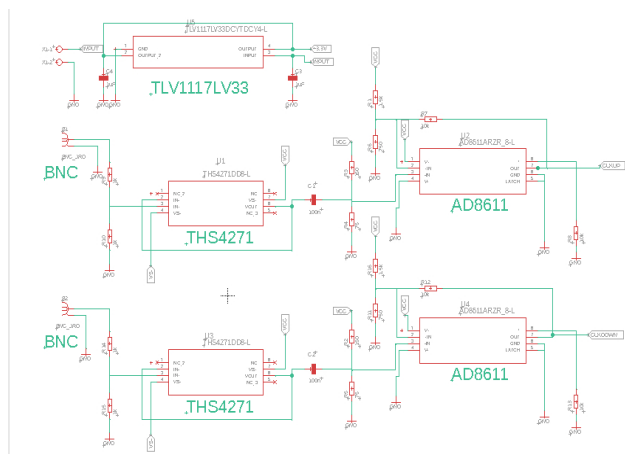


Figura 5: Esquemático de la etapa de comparación

Observación: Para las pruebas experimentales se usó una placa ya existente en el Radio Observatorio de Jicamarca que cumplía la misma función de transformar ondas sinusoidales a cuadradas.

4.2 Etapa del divisor de frecuencia y detector de Fase

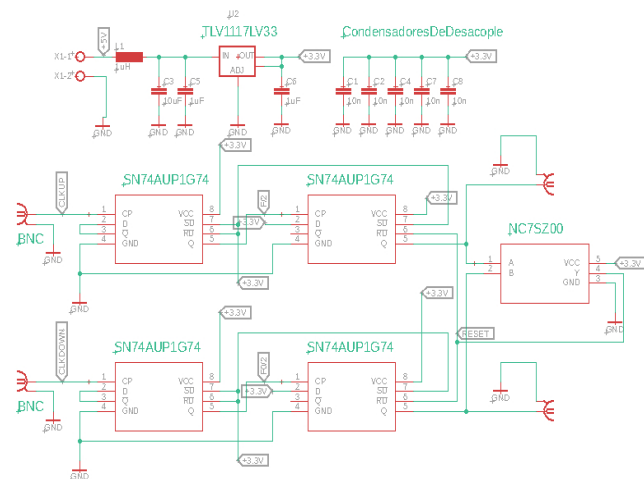


Figura 6: Esquemático de la etapa de divisor de frecuencia y detector de fase

4.3 Etapa del Filtro Pasa Bajo

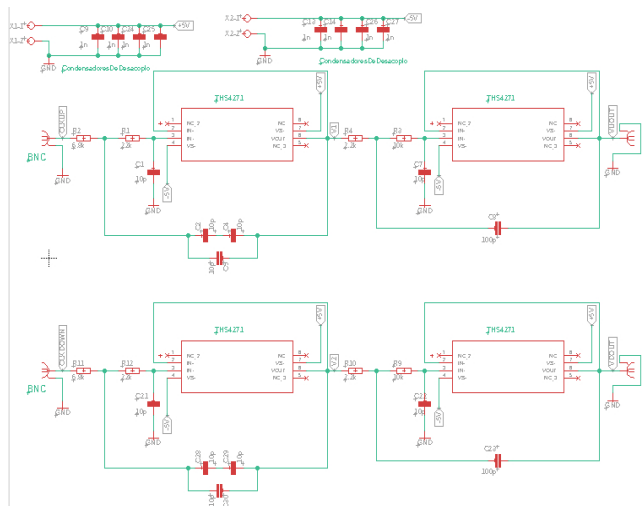


Figura 7: Esquemático de la etapa de filtro pasa bajo

5. RESULTADOS

5.1. Simulación en LTspice

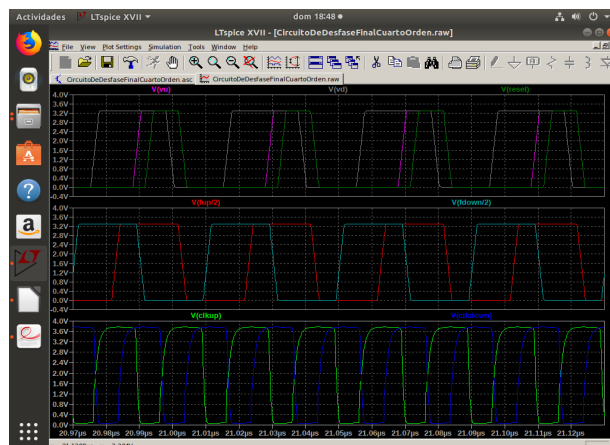


Figura 8: Señales internas del diseño a un desfase de 135°

En la figura 8 el Vckup y Vckdown son las señales producidas después de la etapa del comparador por Histéresis. En la teoría sería: $Vckup = Vi$ y $Vckdown = Vo$. La señal $V(fup/2)$ es la señal que se origina a la salida de la etapa del divisor de frecuencia por el flip flop de arriba y $V(fdown/2)$ es la señal que se origina a la salida de la etapa del divisor de frecuencia por el flip flop de abajo. La señal $V(vu)$ es la señal que se origina en una salida de la etapa del detector de frecuencias. Esta señal vendría a ser en este ejemplo los pulsos. La señal $V(vd)$ es la señal que produce por la otra salida del detector de frecuencias. Esta señal en este ejemplo es la señal cuadrada. La señal $V(reset)$ es la señal que se origina a la salida del AND.

5.2. Resultados Experimentales

5.2.1. Respuesta en el osciloscopio para un desfase de 5 grados a la salida del detector de fase

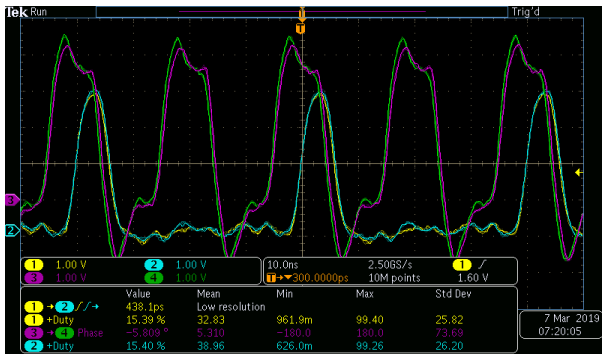


Figura 9: Respuesta de dos señales con un desfase de 5°

Las señales verde y morada representan las señales de entrada desfasadas 5° luego de pasar por el comparador con Histéresis. Las señales amarilla y turquesa representan las señales de salida después de la etapa de desfase (Vuout y Vdout).

5.2.2. Ploteo del Voltaje Vd vs. Desfase

Se halló el ploteo de $V_d = (V_{uout} - V_{dout})$ vs. Desfase, para los ángulos: 0°, 1°, 19°, 40°, 105°, 120°, 125°, 167°, 217°, 220°, 240°, 330 y 356° experimentalmente con las placas diseñadas y tenemos

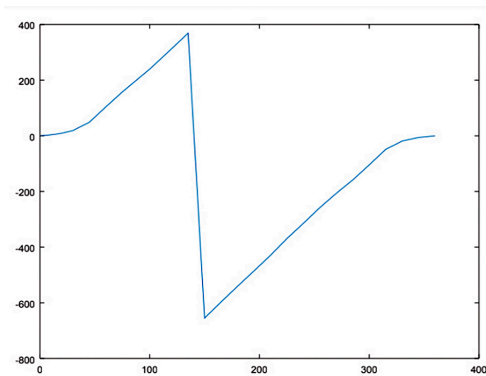


Figura 10: Ploteo de Vd vs Desfase

6. DISCUSIÓN

Con respecto a las simulaciones, en la figura 8 se muestra el comportamiento de las señales internas, los cuales cumplen con lo esperado en la teoría.

Con respecto a los resultados físicos en el osciloscopio, la figura 9 nos da la respuesta de dos señales desfasadas 5 grados. Las señales de salida al ser tan pequeñas tienden a tener una forma de curva en su parte más alta en vez de cuadradas. Se comprobará si es correcto el valor de voltaje de estas señales cuando tengamos el valor de corriente continua en la etapa de filtro,

Con respecto al ploteo de la figura 10, vemos que la gráfica cumple con la tendencia en forma de la gráfica teórica.

7. CONCLUSIONES

De la simulación podemos ver que las señales internas cumplen con el comportamiento deseado.

Se logró comprobar que la gráfica de ploteo experimental cumple con la tendencia teórica a alta frecuencias.

Al hallar los parámetros que describen la ecuación de la gráfica ploteada sería posible hallar cualquier desfase teniendo como dato solo los valores de voltajes en continua a las salidas de los filtros (VUOUT y VDOUT).

8. RECOMENDACIONES

Buscar la simetría en el diseño de la placa de los filtros y el detector de fase para no inducir un desfase por diseño.

Las medidas de las longitudes de las pistas a la salida del detector de fase en el Board del Eagle deben ser iguales para no inducir algún desfase por diseño.

Verificar que los componentes que te otorgan del almacén son los correspondientes a los solicitados.

9. AGRADECIMIENTOS

Al instituto Geofísico del Perú por darme la oportunidad de desarrollarme en mi profesión colaborando y aportando a la ciencia e ingeniería.

A mi asesor, el Ing. Joaquín Verástegui por su guía, paciencia y enseñanzas en el campo de la ingeniería y diseño electrónico.

10. BIBLIOGRAFÍA

Phase Locked Loope - Circuit Design. Autor: Dan. H Wolaver

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA ELECTRO-MECÁNICO DE RADAR DE APERTURA SINTÉTICA CON CONTROL DE APUNTE PARA MODOS STRIPMAP, SPOTLIGHT Y SCAN

**Design and implementation of a synthetic aperture radar electro-mechanical system with automatic antenna pointing for stripmap, spotlight, and scan modes*

Camilo Daniel Ortecho Huamanchumo

Universidad Nacional Federico Villarreal
Facultad de Ingeniería Electrónica e Informática
daniel.ortecho@jro.igp.gob.pe,
daniel.ortecho.jro@gmail.com

Asesor: Dr. Marco Milla Bravo

Área CIELO - Radio Observatorio de Jicamarca
- IGP

RESUMEN

El deshielo de la capa de nieve de la Cordillera de los Andes es un resultado negativo del cambio climático, que ocasiona la pérdida de recursos hídricos para la vida humana.

Motivados por esta situación, los científicos están estudiando la variación de nieve mediante campañas en los Andes. Con el fin de contribuir con este tipo de mediciones, el Instituto Geofísico del Perú está desarrollando en el Radio Observatorio de Jicamarca un sistema de Radar de Apertura Sintética que será utilizado para medir la variación de nieve seca en las montañas nevadas del Perú. El sistema trabajará en banda C (4-7 GHz) y tendrá la capacidad de operar en tres diferentes modos, tales como: stripmap, spotlight y scan.

Palabras clave: Apertura Sintética, Stripmap, Spotlight, Scan, Antena, Radio Observatorio de Jicamarca.

ABSTRACT

Melting of snow covering the Andes Mountains is a negative result of climate change, which cause the loss of water resources for human life. Motivated by this situation, scientists are studying the variation of snow throughout seasons in the Andes. In order to contribute with this type of measurements, the Geophysical Institute of Peru is developing at the Jicamarca Radio Observatory a Synthetic Aperture Radar (SAR) system that would be used to measure the variation of dry snow in the snow-capped mountains in Peru. The system will work at C-band (4-7 GHz) and it will be able to operate in three different modes: stripmap, spotlight, and scan.

Keywords: Synthetic Aperture, Stripmap, Spotlight, Scan, Antenna, Jicamarca Radio Observatory.

1. INTRODUCCIÓN

En el Perú, la Cordillera Blanca, cadena de montañas localizado en Huaraz, Ancash, ha sido impactado por el cambio climático. Entre sus más conocidas montañas con cobertura de nieve se encuentra Pastoruri. (Fig. 1)

Para monitorear la cantidad de variación de nieve, distintas técnicas de medición han sido implementados, tales como, estacas de madera [1] y radares de penetración de tierra [2] sin embargo, las técnicas mencionadas necesitan intervención humana en el nevado. Otros métodos, tales como, escáneres láseres [3] y vehículos aéreos no tripulados [4], necesitan condiciones de clima despejado para óptimas mediciones.

Para superar las dificultades mencionadas anteriormente, los sistemas de radar de apertura sintética (con sus siglas en inglés Synthetic Aperture Radar - SAR) han surgido como una solución innovadora, ellos pueden ser usados para sensar largas áreas de terreno de forma remota, sin necesidad que el operador se encuentre en el lugar del sistema.

El objetivo de esta tesis es diseñar e implementar un sistema SAR que medirá variaciones de nieve de forma remota y bajo cualquier tipo de clima.

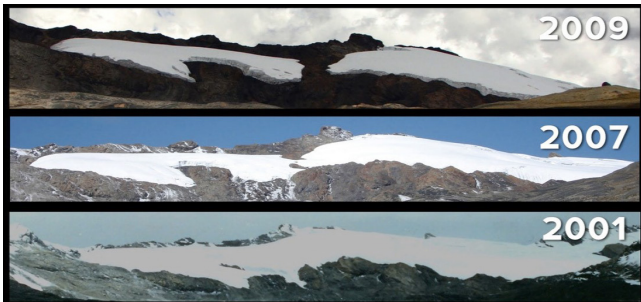


Fig. 1: Variación de nieve del Glaciar Pastoruri al paso de las décadas. [5]

2. DISEÑO DE LA ANTENA

Uno de los principales componentes de un sistema de radar de apertura sintética es la antena. Se usará antenas de bocina que operen en la banda de frecuencias de microondas, (alrededor de 5.3 GHz) para medir nieve seca. [6]

A. Requerimientos de Diseño

La antena de bocina está compuesta de dos partes, su guía de onda y su bocina. En este proyecto, se usará las dimensiones de la guía de onda rectangular tomados del catálogo de dispositivos estándares de RF (Compañía Ainfoinc). [7]

Para diseñar la bocina, es necesario tener en cuenta diferentes parámetros, tales como:

- Frecuencia Central
- Dimensiones de la guía de Onda
- Ganancia de Antena

B. Ecuaciones de Diseño

Basado en los parámetros mencionados anteriormente, realizamos el cálculo de las dimensiones para la bocina con las siguientes ecuaciones. [8]

$$G_0 = 10 \log(g) \quad (1)$$

$$\lambda = c/f \quad (2)$$

$$\chi = g/2\pi\sqrt{2\pi} \quad (3)$$

$$(\sqrt{2\chi} - \frac{b}{\lambda})^2 (2\chi - 1) = \left(\frac{G_0}{2\pi} \sqrt{\frac{3}{2\pi} \frac{1}{\chi} - \frac{a}{\lambda}}\right)^2 \left(\frac{G_0^2}{6\pi^3 \chi} - 1\right) \quad (4)$$

$$\rho_e = \lambda \cdot \chi \quad (5)$$

$$\rho_h = \frac{g^2}{8\pi^3} \left(\frac{\lambda}{\chi}\right) \quad (6)$$

$$a_1 = \sqrt{3\lambda\rho_h} \quad (7)$$

$$b_1 = \sqrt{2\lambda\rho_e} \quad (8)$$

$$P_e = (b_1 - b) \sqrt{\left(\frac{\rho_e}{b_1}\right)^2 - \frac{1}{4}} \quad (9)$$

$$P_h = (a_1 - a) \sqrt{\left(\frac{\rho_h}{a_1}\right)^2 - \frac{1}{4}} \quad (10)$$

C. Procedimiento de Diseño

Basado en las ecuaciones presentadas con anterioridad, se calcula las dimensiones de bocina usando el siguiente procedimiento.

1. Como primer punto, la ganancia (G_0) en dB se convierte a ganancia (g) en escala lineal usando la ecuación (1).
2. Luego, la longitud de onda (λ) para $f=5.3$ GHz es calculada usando la ecuación (2), donde c es la velocidad de la luz.
3. El coeficiente de calibración (χ) se encuentra operando la ecuación (3). A su vez, se debe verificar que χ cumpla la ecuación (4).
4. Las distancias a lo largo de la superficie de la antena (ρ_e y ρ_h) son encontradas usando las ecuaciones (5) y (6), respectivamente.
5. Los parámetros de la apertura de la bocina (a_1 y b_1) se estiman de las ecuaciones (7) y (8) respectivamente.
6. Finalmente, los parámetros P_e y P_h se encuentran empleando las ecuaciones (9) y (10). Si el diseño es correcto estos parámetros deberán ser iguales.

En la Fig. 2, se muestra la antena de bocina y la ubicación de los parámetros eléctricos.

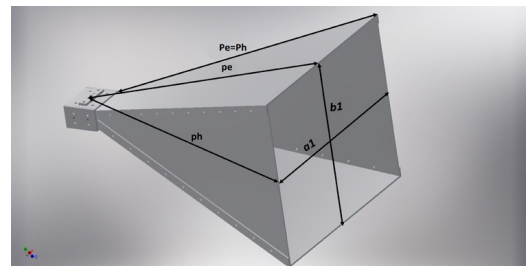


Fig. 2: Parámetros de la antena de bocina

3. DISEÑO DEL SISTEMA MECÁNICO

El sistema mecánico está compuesto por dos partes, el sistema mecánico de control de posición que controla el movimiento longitudinal de las antenas y el sistema de soporte mecánico que sostiene las antenas de bocina.

A. Sistema mecánico de control de posición

Para el mecanismo de posición longitudinal se puede usar distintos tipos de transmisión mecánica, tales como, husillo de bolas, piñón y cremallera, faja transportadora, entre otros.

Los más usados para aplicaciones de dimensión pequeña son la faja transportadora y el husillo de bolas. Además, ambos mecanismos son de bajo costo de implementación. La tabla I muestra la comparación entre estos dos tipos de mecanismos. [9]

Parámetros	Husillo de bolas	Faja transportadora
Resistencia a la suciedad	Alto	Alto
Facilidad de Instalación	Alto	Medio
Precisión	Alto	Alto
Fuerza de carga	Alto	Medio
Mantenimiento	Medio	Medio
Factor de Ruido	Medio	Bajo

Tabla I: Tipos de transmisión mecánica

B. Sistema mecánico de soporte

El sistema mecánico de soporte debe sostener las antenas de bocina y la instrumentación electrónica (componentes de radar). Por lo tanto, este sistema debe soportar, como mínimo, dos veces el peso de los componentes del radar.

El criterio máximo de energía de distorsión (Teoría de Huber - Mises - Hencky) establece que el límite de elasticidad es alcanzado cuando la energía de distorsión alcanza un valor crítico. La Tensión de Von Mises es proporcional a la energía de distorsión.

En la ecuación (11) se describe el Esfuerzo de Von Mises (σ_f) en términos de los esfuerzos principales (σ_1, σ_2 y σ_3).

$$\sigma_f = \sqrt{\frac{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_1 - \sigma_3)^2}{2}} \quad (11)$$

De igual manera, otro parámetro importante es la deformación o desplazamiento del material sometido a esfuerzos. Este parámetro cuantifica el efecto de la carga en la pieza. Para encontrar el desplazamiento vertical ($\delta^2 v / \delta x^2$), la ecuación (12) de la curva elástica es aplicada, [11]

$$\frac{\delta^2 v}{\delta x^2} = \frac{M_z(x)}{E \cdot I_z} \quad (12)$$

Donde:

$M_z(x)$ = Momento flexionante,

I_z = Momento de Inercia,

E = Módulo de Young.

El diseño mecánico en ingeniería establece que cualquier mecanismo que está sujeto a una fuerza debe tener un factor de seguridad. De este modo, el mecanismo trabajará convenientemente sin riesgo de deformación permanente. En la ecuación (13) se describe el factor de seguridad (N_{ult}) en términos de la fuerza máxima de carga.

$$\sigma_{ult} = \sigma_w \cdot N_{ult} \quad (13)$$

Donde:

σ_{ult} = Esfuerzo admisible,

σ_w = Esfuerzo de trabajo.

4. SISTEMA DE CONTROL

El modelo matemático del sistema se obtuvo de manera experimental a partir de la curva de reacción excitado por una señal escalón.

En la Fig. 3 se muestra la curva de reacción de velocidad del motor excitado por una señal escalón.

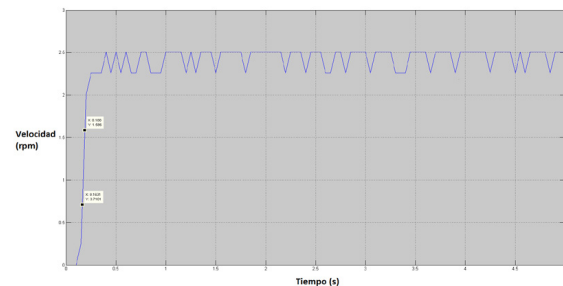


Fig. 3: Curva de reacción del sistema.

Mediante métodos de identificación de sistemas [12] se encuentra la función de transferencia de velocidad del motor.

$$\frac{\omega(s)}{V(s)} = \frac{0.502}{0.03s + 1}$$

Debido a que se requiere controlar la posición, según la teoría de control se le aplica un integrador en el dominio Laplaciano (1/S).

A continuación, se muestra la función de transferencia de posición (FT(θ)) del motor.

$$\frac{\theta(s)}{V(s)} = \frac{0.502}{0.03s^2 + s}$$

Sin embargo, para realizar la programación del microcontrolador para el control angular de las antenas se necesita que la función de transferencia se encuentre en el plano discreto de Z. A continuación, se presenta la (F.T(θ)) discreta.

$$\frac{\theta(z)}{V(z)} = \frac{0.000198z + 0.0001873}{z^2 - 1.847z + 0.8466}$$

5. RESULTADOS

La fabricación de las antenas de bocina fue realizada usando planchas de aluminio y remaches. La tabla II y la tabla III muestran los parámetros de diseño y los parámetros de la bocina.

Parámetros	Dimensiones
Frecuencia Central	5.3 GHz
Ganancia de la antena	20 dB
Dimensión de la guía de onda	a= 4.039 cm (largo) b= 2.019 cm (ancho) c= 3.8 cm (altura)

Tabla II: Parámetros de diseño

Parámetros	Dimensiones
χ	6.63
ρ_e	37.54 cm
ρ_h	34.40 cm
a_l	24.17 cm
b_l	20.61 cm
P_e	29.61 cm
P_h	29.61 cm

Tabla III: Parámetros de bocina

En las siguientes figuras, se puede ver las diferencias entre el dipolo en espacio libre (Fig. 4a) y el dipolo con el chasis (Fig. 4b), fijados a la misma frecuencia central (5.3 GHz), es considerable (17.8 dB). Por lo tanto, el primer prototipo es funcional.

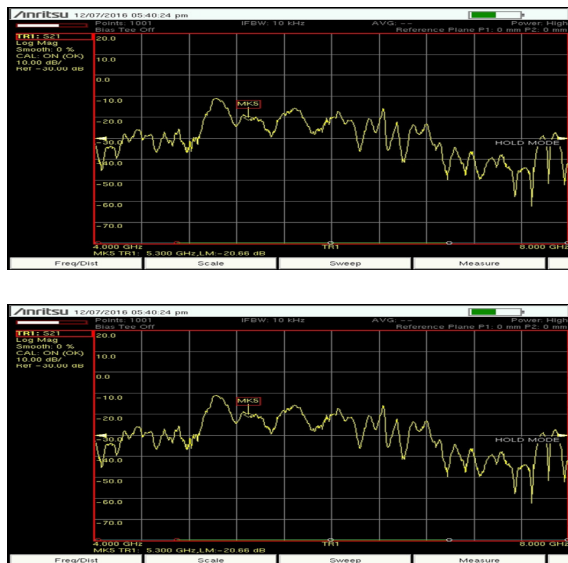


Fig. 4: Ganancia de antena: dipolo sin chasis (4a) y dipolo con chasis (4b).

En la Fig. 5 se muestra el análisis de esfuerzo de la pieza “base antena” sujeta a una fuerza de 57N y un momento de 2.1 N.m, generado por el peso del mecanismo de antena y el actuador, respectivamente.

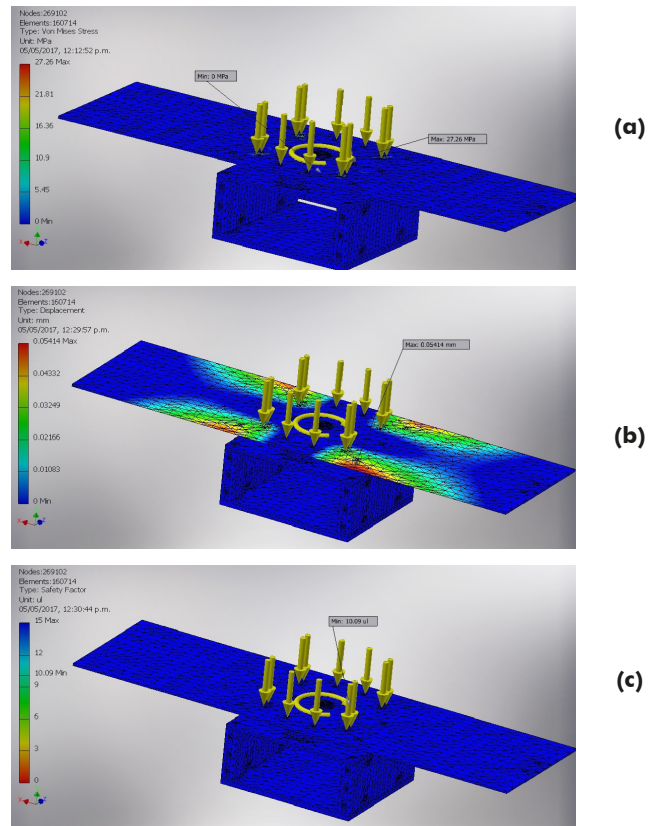


Fig. 5: Análisis de esfuerzos de la pieza “Base Antena” (5a) Tensión de Von Mises (5b) Desplazamiento (5c) Factor de seguridad.

Autodesk Inventor® simula diversos tipos de análisis, tales como, Tensión de Von Mises, Desplazamiento y Factor de Seguridad, basados en la ecuación (11), (12) y (13), respectivamente. La tabla IV muestra estos parámetros.

Tipo de Análisis	Resultados
Tensión de Von Mises	27.26 MPa
Desplazamiento	0.05414 mm
Factor de seguridad	10.09
Máximo esfuerzo	575.13 N

Tabla IV: Resultados de la simulación del análisis de esfuerzo.

En la Fig. 6 se muestra la gráfica de control del modo scan, en la que se observa que se tiene repetibilidad para 5 muestras.

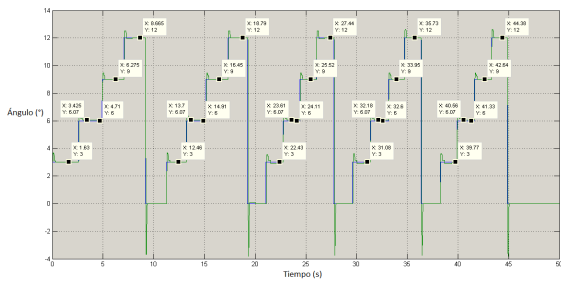


Fig. 6: Modo scan del SAR

En la Fig. 7 se muestra la implementación de la antena con el mecanismo de soporte.

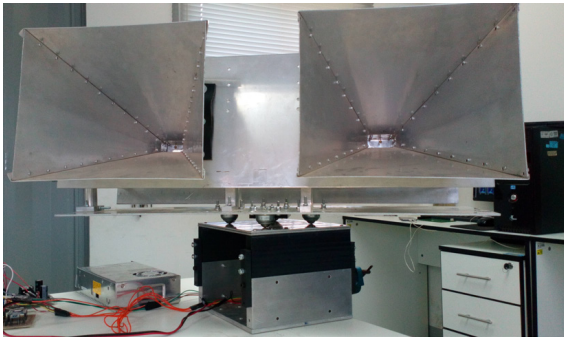


Fig. 5: Antenas de bocina empotradas en el mecanismo de soporte.

6. DISCUSIÓN

Se puede comprobar que los resultados obtenidos al término de la fabricación de las antenas, son mejores que los calculados con el software Feko. A su vez una de los grandes aportes del radar de apertura sintética, es que obtiene mejores mediciones en precisión, y asegura la vida del operador; lo que no se puede con los métodos tradicionales. Una de las limitaciones del sistema, es que el radar debe posicionarse en un punto determinado, sin que se altere las condiciones de este lugar.

El presente trabajo se basa en el análisis, diseño e implementación del sistema mecánico de radar y sus modos de apunte, pero no llega a extenderse al estudio de los datos obtenidos por el sensor; con unas pruebas de campo y el análisis en el procesamiento de la data, así como de sus beneficios de apunte, se llegaría a validar que el radar no solo podría medir manto nevoso (stripmap o scan), sino también a ayudar en la búsqueda de personas que se encuentren atrapadas en la nieve (spotlight).

7. CONCLUSIONES

Este trabajo describe el diseño de la estructura mecánica de un radar de apertura sintética. La banda de frecuencia operacional de las antenas determina el tamaño de las antenas de bocina. La principal aplicación de este radar es medir la variación relativa de nieve seca con respecto a un nivel de referencia de la profundidad actual de nieve.

El análisis de estrés nos da diferentes parámetros, tales como, factor de seguridad, desplazamiento y máximo estrés los cuales nos permiten saber la máxima fuerza que puede ser aplicada a la estructura antes de tener deformaciones permanentes.

Se implementó un sistema de control para el movimiento angular de las antenas que están incluidas en el mecanismo de soporte de las antenas, y se probó el sistema para el modo scan.

En futuros trabajos, se presentará el modo spotlight del radar SAR, ya que para el modo stripmap no se necesita rotar las antenas.

8. RECOMENDACIONES

Para diseñar estructuras mecánicas, se recomienda utilizar los programas Autodesk Inventor o Solidwork, debido a que permiten realizar múltiples análisis en el diseño mecánico de la estructura; simulación dinámica para verificar si existe colisión, análisis de deformación y factor de seguridad de las piezas diseñadas, etc.

9. AGRADECIMIENTOS

Se agradece al equipo CIELO (Cluster of Instruments for Equatorial & Low-Latitude Observations) del Radio Observatorio de Jicamarca, al Dr. Marco Milla Bravo por su apoyo constante en el proyecto. Así como también, al FONDECYT por el apoyo a este proyecto mediante el acuerdo N° -PIAP-2-P-434-14.

El Radio Observatorio de Jicamarca es una instalación del Instituto Geofísico del Perú (IGP) operado con el soporte de NSF Award AGS-1433968 mediante la Universidad Cornell.

10. BIBLIOGRAFÍA

Instituto Nacional de Investigación en glaciares y ecosistemas, Monitoreo Glaciológico en el glaciar Llaca [Online] disponible en http://www.inaigem.gob.pe/download/glaciares/inspeccion_tecnica_monitoreo/it_3.pdf, 2014.

A. Rivera, G. Gacitúa, R. Zamora, J. Uribe y J. Oberreuter, Mediciones de espesor de hielo en Chile usando radio eco sondaje, Valdivia, Chile: Departamento de Geografía. Universidad de Chile y Centro de estudios científicos, 2016.

G. Luzi, L. Noferini, D. Mecatti, G. Macaluso, M. Pieraccini, C. Atzeni, A. Schaffhauser, R. Fromm y T. Nagler, Using a Ground-Based SAR Interferometer and a Terrestrial Laser Scanner to Monitor a Snow Covered Slope: Results From an Experimental Data Collection in Tyrol, Austria, 2009.

E. Porcel, Iniciación de un monitoreo del balance de masa en el glaciar Suyuparina, Cordillera Vilcanota, Perú, 2014.

K. Stanto, Interview with "Glacial Balance" Filmmaker Ethan Steinman [Online] Disponible en https://www.picarro.com/community/blog/interview_with_glacial_balance_filmmaker_ethan_steinman California, 12 de Junio del 2012.

A. Martínez, Snow cover monitoring techniques with GB-SAR Barcelona, España, UPC, 2008.

Ainfoinc, LB-159-20 - 4.90 - 7.05 GHz Standard Gain Horn Antenna, [Online] Disponible en http://www.ainfoinc.com/en/pro_pdf/new_products/antenna/Standard%20Gain%20Horn%20Antenna/tr_LB-159-20.pdf.

C. Balanis, Antenna Theory – Analysis and Design, pp 65-66, 2005

N. Castiglione, Transmisión del movimiento a los ejes de los CNC-ROUTERS, 2011.

R. Von Mises, Mechanik der festen körper im plastisch deformablen Zustand. Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse, pp 582-592, 1913.

Pytel and Al. Singer, Resistencia de Materiales, México, 1987.

L. Lennart. "SYSTEM IDENTIFICATION: Theory for the User ". Prentice-Hall, Inc., 1987.

ENVIO DE DATOS POR GIGABIT ETHERNET DESDE UNA INTERFAZ LVDS USANDO SOC (SINGLE BOARD COMPUTER + FPGA)

**Data transmission through gigabit ethernet from a lvds interface using a soc (single board computer + fpga)*

David George Shatwell Pittaluga

Universidad de Ingeniería y Tecnología - UTEC
Departamento de Ingeniería Electrónica
david.shatwell@utec.edu.pe

Asesor: Joaquín Verástegui

Área de I&D+I – Radio Observatorio de Jicamarca - IGP

RESUMEN

En el presente trabajo se hace una descripción del diseño e implementación de un sistema capaz de transmitir datos a alta velocidad desde JARS 2.0 hacia un PC por medio de Gigabit Ethernet usando un SoC. El sistema consta principalmente de dos partes: adquisición de datos de la interfaz LVDS y envío hacia la PC por medio de un protocolo de comunicación. Para la adquisición de datos de la interfaz LVDS fue necesario diseñar un sistema en el FPGA que demultiplexe y copie los datos a una memoria compartida con el procesador. Luego, para el envío de datos se diseñó un software que lea los datos de la memoria compartida y los envíe hacia la PC por medio del protocolo UDP.

Palabras clave: JARS 2.0, Gigabit Ethernet, SoC, LVDS, UDP

ABSTRACT

The objective of this project is to design and implement a system capable of transmitting data from JARS 2.0 to a remote PC through a high speed Gigabit Ethernet using a SoC. The system has two main parts: data acquisition from the LVDS interface and transmission of the data to the PC through a communication protocol. To acquire the data from the LVDS interface, it was necessary to design a system in the FPGA that multiplexes and copies the data to a memory shared with the processor. Then, a software was designed to read the data from the shared memory and send it to the PC through the UDP protocol.

Keywords: JARS 2.0, Gigabit Ethernet, SoC, LVDS, UDP

1. INTRODUCCIÓN

El sistema JARS adquiere los datos del radar principal y los transmite por medio de una interfaz LVDS. Para que los datos sean recolectados y posteriormente procesados por una computadora, es necesario algún tipo de sistema que adquiera los datos de la interfaz LVDS y los envíe por un protocolo de comunicación estándar. En la versión 2.0 de JARS se usa un FPGA Spartan 6 en el cual se ha implementado la pila del protocolo UDP para poder enviar los datos por Ethernet. Sin embargo, este sistema tiene las siguientes desventajas: la tarjeta de evaluación y el software que se usa para programarla (ISE Design Suite) están descontinuados, el conector LVDS es propietario, el ancho de banda de la comunicación está limitado por el FPGA y el protocolo está limitado únicamente a UDP.

Actualmente existen mejores alternativas como los SoC (System-on-Chip), los cuales poseen un FPGA y un procesador. La principal ventaja de un SoC para la implementación de una interfaz LVDS-Ethernet es que permite el uso de sistemas operativos Linux que manejan toda la carga de la red. Es decir, no es necesario implementar la pila de los protocolos TCP/IP y UDP en el FPGA. Otras ventajas de los SoC son su bajo costo (alrededor de \$100), velocidad elevada (600 MHz), conectores comunes con opción a LVDS y tamaño compacto.

Envío de Datos de JARS 2.0

El sistema JARS envía los datos adquiridos del radar principal como words de 32 bits. Cuando se usan los 8 canales de la antena, JARS tiene la capacidad de enviar datos a una frecuencia de 1 MHz, lo cual representa una velocidad de 256 Mbps.

DE0-Nano-SoC

El DE0-Nano-SoC es una tarjeta de desarrollo de Terasic que contiene un SoC Cyclone V de Intel (antiguamente Altera). El Cyclone V está compuesto por un FPGA y un procesador ARM A9 de dos núcleos. La comunicación entre el FPGA y el procesador se da a través de tres buses AXI, uno de control (32 bits) y dos de datos (hasta 128 bits).

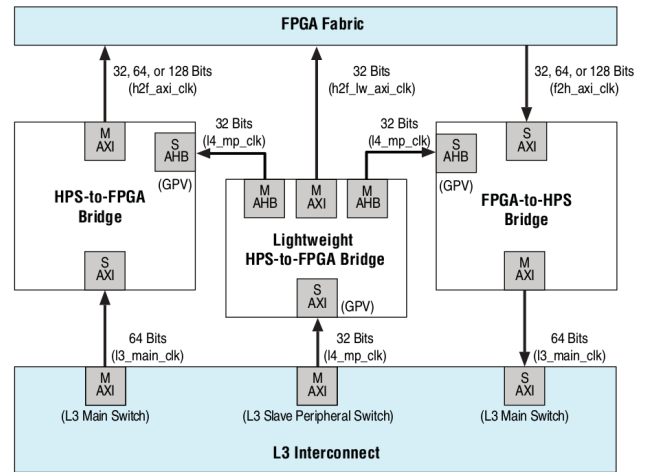


Figura 1. Sistema de comunicación HPS-FPGA

DMA

Un DMA (Direct Memory Access) es un circuito que permite a los periféricos acceder a la memoria principal del sistema independientemente del CPU. Sin un DMA, el CPU está totalmente ocupado durante las operaciones de lectura/escritura y por lo tanto, no puede realizar otras operaciones en paralelo.

2. DESARROLLO

Para desarrollar el sistema del SoC, lo primero que se establece es el flujo de datos del FPGA al procesador. La figura 2 muestra el sistema propuesto, donde las flechas delgadas son señales de control y las gruesas el flujo de datos.

Con el flujo de datos establecido, se puede separar el diseño del sistema en tres partes: creación de bloques del FPGA escritos en VHDL y Verilog, instanciación de IP cores y conexiones entre el FPGA y el procesador con la herramienta Platform Designer y la creación del software en lenguaje C.

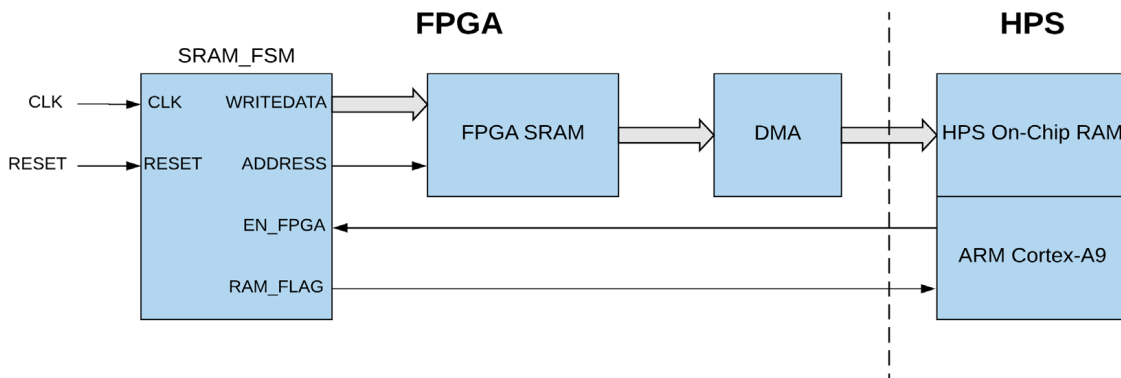


Figura 2. Flujo de datos del sistema

2.1. FPGA

En el FPGA se creó una máquina de estados llamada SRAM_FSM cuya función es generar una secuencia de datos y direcciones de memoria para la SRAM a una frecuencia de 8 MHz.

Cabe resaltar que, en el sistema real, los datos del JARS llegan en paquetes de 8 bytes y deben ser demultiplexados antes de ser guardados en memoria. Sin embargo, para hacer pruebas resulta más fácil usar un generador de números conectado directamente a la memoria SRAM.

2.2. SoC

En la parte del SoC, se instanciaron los siguientes IP cores de Intel.

2.2.1. PLL

Se usó un PLL que genera 2 señales de reloj a partir de una referencia de 50 MHz. El primer reloj generado es de 100 MHz y está conectado a los buses HPS-FPGA, la SRAM y a los PIO. El segundo reloj es de 400 MHz y se usa exclusivamente para que

el bloque "NUM_GENERATOR" pueda generar números a una velocidad de 8 MHz.

2.2.2. HPS

El bloque HPS (Hard Processor System) se refiere al procesador ARM Cortex-A9. A pesar de la gran cantidad de entradas y salidas que tiene, el procesador solo utiliza dos de ellas. El bus "h2f_axi_slave" está conectado al puerto "write_master" del DMA y sirve para adquirir los datos del FPGA. Por otro lado, el bus "f2h_axi_master" está conectado al puerto de control del DMA y a los PIO "en_fpga" y "sram_flag", las cuales sirven como señales de control entre el FPGA y el procesador.

2.2.3. DMA

El bloque DMA se usa para copiar los datos de la memoria SRAM en el FPGA a la memoria On-Chip RAM del procesador. Este bloque lee los datos de la SRAM por el puerto "read_master" y escribe en la On-Chip RAM por el puerto "write_master". Además, tiene un puerto controlado por software que se usa para indicar las posiciones de memoria que se van a transferir del FPGA al procesador.

2.2.4. SRAM

El bloque SRAM es una memoria de 64 KB con words de 32 bits instanciada en el FPGA. Esta memoria tiene dos puertos esclavos: uno para lectura y otro para escritura. Como se mencionó anteriormente, el puerto de lectura está conectado al DMA mientras que el puerto de escritura es exportado al FPGA y controlado por el bloque "SRAM_FSM".

2.2.5. PIO: Habilitador de FPGA y Bandera de SRAM

El bloque "en_fpga" funciona como una señal de control de 1 bit que se usa para habilitar los bloques del FPGA desde el programa principal. Su puerto "s1" está conectada al puerto "h2f_axi_master" del procesador.

Por otro lado, el bloque "RAM_flag" funciona como una señal de control de 6 bits que se usa para indicarle al procesador qué sección de la memoria SRAM está siendo escrita. Su puerto "s1" también está conectada al puerto "h2f_axi_master" del procesador.

2.3. Procesador ARM Cortex-A9

Para recolectar los datos de la SRAM y enviarlos por Ethernet se escribió un programa en C cuyo diagrama de flujo se muestra en la figura 3.

2.3.1. Adquisición de Direcciones de Memoria Virtual

Lo primero que se debe hacer en el programa es definir las macros con las direcciones de memoria a la cual están mapeados los componentes del FPGA (tabla 1). Debido a que Linux no permite al usuario acceder directamente a la memoria física, también se deben definir punteros que almacenen las direcciones virtuales de memoria.

Luego, dentro del función principal se abre el archivo /dev/mem para obtener un descriptor de archivo y se adquiere la dirección de memoria virtual del bus LW HPS-to-FPGA Bridge con la función mmap(). Los punteros que apuntan a la dirección de memoria virtual del puerto de control del DMA y los componentes "sram_flag" y "en_fpga" se definen como la dirección del bus sumados con un desfase. En este caso, el desfase de memoria de los componentes se muestra en la tabla 1.

Componente	Nombre del puerto	Dirección base	Desfase
F2H AXI Slave	hps_0.f2h_axi_slave	0xFFFF0000	0x00000000
DMA Control Port	dma.control_port_slave	0xC0000000	0x00000000
SRAM	SRAM_0.s1	0xC0000000	0x00000000
Habilitador de FPGA	en_fpga.s1	0xC0000000	0x00000020
Bandera SRAM	sram_flag.s1	0xC0000000	0x00000030

Tabla 1. Direcciones de memoria virtual usadas en el programa

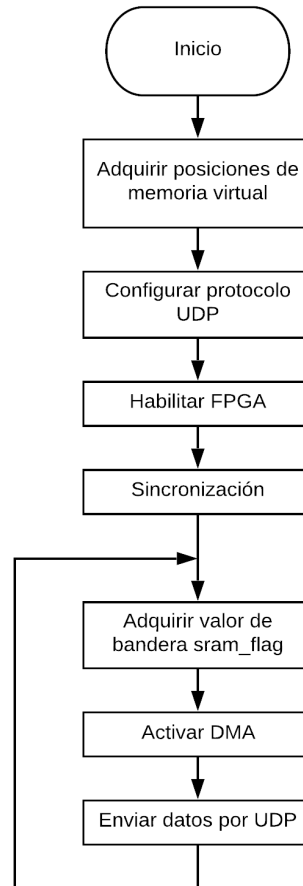


Figura 3. Diagrama de flujo del programa del SoC

2.3.2. Configuración de UDP

Para hacer las configuraciones de UDP lo primero que se necesita es abrir un socket para transmitir datos (Tx) con la función socket() y especificar que se va a usar IPv4 y el protocolo UDP. Además, se crea una estructura "talker" con la dirección IP y el puerto al cual se envían los datos.

2.3.3. Sincronización

Antes del lazo principal, se utiliza la señal de control "en_fpga" para habilitar el FPGA de modo que el primer dato que llega se encuentre en la posición de memoria uno. Luego, el programa espera a que suceda un cambio en la bandera "sram_flag" para que el DMA empiece a transferir los datos hacia la On-Chip RAM. Este proceso de sincronización asegura que en todo momento se ejecuten los siguientes tres procesos en paralelo:

- El procesador envía por UDP los datos en la sección "n" de la memoria On-Chip RAM.

- El DMA copia la sección “n+1” de la memoria SRAM a la memoria On-Chip RAM.
- La máquina de estados guarda los datos que vienen del JARS en la sección “n+2” de la memoria SRAM.

2.3.4. Lazo principal

En el lazo principal, lo primero que se hace leer nuevamente la bandera de la SRAM y empezar a transferir la sección de memoria correspondiente con el DMA. A la misma vez que se transfieren los datos con el DMA, el programa determina las direcciones de memoria necesarias de la sección “n” y utiliza la función sendto() para enviar los datos de dicha sección por UDP. La función sendto() tiene como argumentos el descriptor de archivos del puerto, el puntero que apunta hacia el arreglo la estructura “server” que se creó anteriormente.

2.4. PC de Destino

Para adquirir los datos enviados por UDP desde el SoC, también se creo que un programa en C similar al del servidor UDP. El diagrama de flujo del programa se muestra en la figura 4.

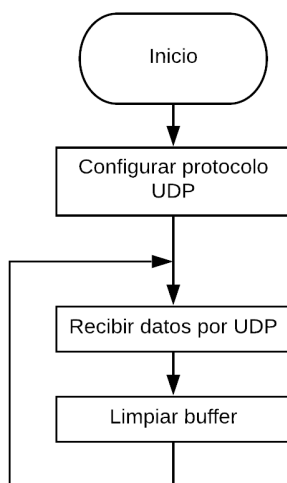


Figura 4. Diagrama de flujo del programa de la PC

2.4.1. Configuración de UDP

Para hacer las configuraciones de UDP lo primero que se necesita es abrir un socket para recibir datos (Rx) con la función socket() y especificar que se va a usar IPv4 y el protocolo UDP. Además, se crea una estructura “listener” con la dirección IP y el puerto por el cual se reciben los datos y un buffer de 1024 bytes para almacenarlos.

2.4.2. Lazo Principal

En el lazo principal del cliente UDP lo único que se hace recibir paquetes con la función receivefrom(), la cual tiene como argumentos el descriptor de archivo generado por la función socket(), el buffer para almacenar los datos y la estructura “listener” creada anteriormente. Una vez que recibe el paquete, se limpia el buffer y se repite el proceso.

3. RESULTADOS

3.1. Validación del Hardware

Para verificar el funcionamiento de la máquina de estados creada en el FPGA, se hizo un testbench con la herramienta Modelsim. Los resultados del testbench se muestran en la figura 5.

Se puede observar que, con una señal de reloj de 350 MHz, se genera un dato y una dirección de memoria cada 125 ns (8 MHz). Esto demuestra que la máquina de estados conectada a la memoria SRAM funciona correctamente.

3.2. Validación del Software

3.2.1. Validación del Contenido de los Paquetes

En la primera prueba que se hizo para validar el software, se verificó que el contenido de los paquetes recibidos por la PC sea igual a aquellos enviados por el SoC. Para ello, se hizo algunas modificaciones al código del SoC y de la PC.

En el SoC, se creó un vector para almacenar el contenido de 10 paquetes consecutivos, lo cual equivale al contenido total de la memoria. Una vez que el lazo principal se ejecuta 10 veces, el programa imprime el contenido del vector junto con el estado de la bandera de la SRAM en la terminal.

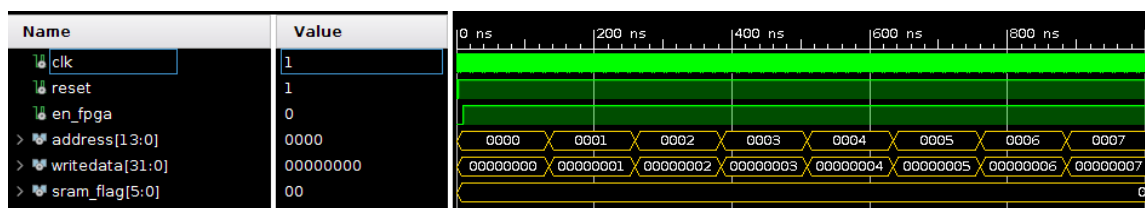


Figura 5. Simulación de la máquina de estados SRAM_FPSM

A partir de esta prueba se determinó que el contenido de todos los paquetes en la PC de destino es idéntico a aquellos mandados por el SoC.

3.2.2. Validación del Porcentaje de Paquetes Recibidos

El objetivo de la segunda prueba fue verificar el porcentaje de paquetes recibidos por la PC. Para ello, se hizo 10 pruebas que consistían en que el SoC envíe 100 paquetes y, con un contador, verificar cuántos paquetes llegan a la PC de destino.

Con esta prueba se determinó que el porcentaje de paquetes recibidos por la PC es de aproximadamente 96.5%.

4. DISCUSIÓN

Para lograr que el sistema de envío de datos funcione correctamente, es necesario tener en cuenta la frecuencia de reloj conectada a los buses de datos y componentes del sistema, especialmente del DMA. Una frecuencia muy baja (<50 MHz) hace que la transferencia de datos entre el FPGA y el procesador sea lento, lo cual a su vez hace que se pierdan paquetes, mientras

que una frecuencia muy alta (> 150 MHz) hace que el procesador del SoC se cuelgue.

En cuanto al porcentaje de paquetes recibidos, se pudo notar que no hubo variación entre las frecuencias intermedias (50 a 100 Mhz). Se cree que esto sucede debido a que el cuello de botella no se encuentra en los buses AXI o en el DMA, sino en el programa del SoC que envía los datos por UDP.

5. CONCLUSIONES

En el desarrollo de este proyecto se logró trabajar con la tarjeta de desarrollo DE0-Nano-SoC para enviar datos generados en el FPGA por Ethernet a una velocidad de 256 Mbps. El sistema que se probó escribe los datos que llegan al SoC en una memoria SRAM, para luego leer y enviar dichos datos por Ethernet.

El uso de la herramienta Platform Designer del programa Quartus Prime resultó ser muy útil para el desarrollo del proyecto. Esta herramienta no solo es necesaria para diseñar los buses que conectan el FPGA con el procesador, sino también para instanciar IP cores de manera sencilla en el FPGA. Sin embargo, para poder crear los buses es necesario entender bien cómo funciona el flujo de datos dentro del SoC. Para ello resultó muy útil leer el manual del Cyclone V, el cual explica de manera detallada cómo funciona el SoC.

6. RECOMENDACIONES

Como trabajo futuro se recomienda buscar la manera de optimizar el código del SoC para poder mejorar el porcentaje de paquetes recibidos. Además, se recomienda realizar un shield para el DE0-Nano-SoC donde se pueda conectar los cables LVDS del JARS y hacer pruebas más exactas.

7. AGRADECIMIENTOS

Agradezco al Instituto Geofísico del Perú a través del Radio Observatorio de Jicamarca por haberme brindado la facilidad de usar los equipos necesarios para este proyecto, el cual me ayudó mucho en mi formación académica y profesional.

Así mismo, quiero agradecer especialmente a mi asesor Joaquín Verástegui y a John Rojas por su apoyo y orientación a lo largo del proyecto.

8. BIBLIOGRAFÍA

(2012). Cyclone V Device Handbook, Volume 3: Hard Processor System Technical Reference Manual. Editado por Altera Corporation, San Jose, California.

(2015). DE0-Nano-SoC User Manual. Editado por Terasic.

Land, B. (2019). DE1-SoC: ARM HPS and FPGA, Addresses and Communication, Cornell ECE 5760. [En línea]. Disponible en https://people.ece.cornell.edu/land/courses/ece5760/DE1_SOC/HPS_peripherals/FPGA_addr_index.html

ESTUDIO DE EFECTOS IONOSFÉRICOS EN LAS SEÑALES DE RADIOFRECUENCIA MEDIANTE EL ANÁLISIS DE DATOS DE RECEPTORES GNSS

**Study of ionospheric effects study on radio frequency signals through the analysis of gnss receiver data*

George Steve Fajardo Soria

Universidad Nacional de Ingeniería
Facultad de Ingeniería Mecánica
george.fajardo.s@uni.pe

Asesor: Dr. Edgardo Pacheco

Área de IDI - Radio Observatorio de Jicamarca - IGP

RESUMEN

Los sistemas mundiales de navegación por satélite (GNSS) como GPS, desempeñan un papel importante en los sistemas modernos de comunicación, navegación, posicionamiento y temporización. Las señales GNSS a menudo se interrumpen al pasar a través de la ionósfera terrestre, que es una región volátil de partículas cargadas que se ven afectadas continuamente por tormentas solares. Este trabajo pretende utilizar información sobre lo que afecta el comportamiento ionosférico, desde el Sol hasta la magnetósfera, combinado con enfoques de machine learning para correlacionar las perturbaciones de la señal GNSS con los fenómenos anteriormente mencionados en la zona ecuatorial magnética.

Palabras clave: GNSS, cintilaciones, aprendizaje automático.

ABSTRACT

Global satellite navigation systems (GNSS), such as GPS, play an important role in modern communication, navigation, positioning, and timing systems. GNSS signals are often interrupted by passing through the Earth's ionosphere, which is a volatile region of charged particles that are continually affected by solar storms. This paper intends to use information on what affects ionospheric behavior, from the Sun to the magnetosphere, combined with machine learning approaches to correlate the disturbances of the GNSS signal with the phenomena mentioned above in the magnetic equatorial zone.

Keywords: GNSS, scintillations, machine learning.

1. INTRODUCCIÓN

La ionósfera es parte de la atmósfera superior de la Tierra, entre 80 y aproximadamente 1000 km, donde la radiación ultravioleta extrema (EUV) y la radiación solar de rayos X ionizan los átomos y las moléculas creando así una capa de electrones. La ionósfera es importante porque refleja y modifica las ondas de radio utilizadas para la comunicación y la navegación. Otros fenómenos como las partículas energéticas cargadas y los rayos cósmicos también tienen un efecto ionizante y pueden contribuir a la ionósfera.

Los átomos y las moléculas atmosféricas se ven afectados por la alta energía de los fotones de rayos X y EUV del sol. Dado que la mayor cantidad de ionización es causada por la irradiación solar, el lado nocturno de la tierra y el polo apuntado lejos del sol (dependiendo de la estación) tienen mucha menos ionización que el lado del día de la tierra y el polo apuntado hacia el sol. (<https://www.swpc.noaa.gov/phenomena/ionosphere>)

Estos fenómenos tales como las irregularidades en la región F

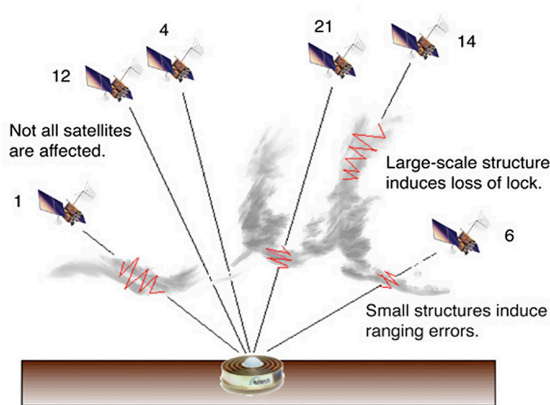


Figura 1. Modelo de errores ionosféricos. Para los satélites 12 y 4 se ve que no son afectados. En el satélite 14 se ve un gran alteración puede causar una pérdida de señal. Para los satélites 1, 21 y 6 se observa pequeñas irregularidades que generan errores.

sobre los 200 km de altitud conocidas como F o el electrochorro ecuatorial, que es una corriente de iones que se produce en la ionósfera, afectan a las señales de radiofrecuencia que atraviesan esta región espacial, como se puede observar en la Figura 1.

El Perú al encontrarse en una zona de baja latitud se encuentra cerca al ecuador magnético, es por ello que es afectado altamente y por ello es importante el estudio de estos fenómenos.

Un índice utilizado para medir el cambio de fase y amplitud en las señales de radiofrecuencia en la ionosfera es el índice S4, mediante este índice podemos caracterizar la ocurrencia de cintilaciones.

Trabajos anteriores como el realizado por Linty, Nicola & Farasin, Alessandro & Favenza, Alfredo & Dosis, Fabio. (2018) o Y. Jiao, J. J. Hall and Y. T. Morton (2017) presentan correlaciones entre diversas variables, usando la potencia de entrada y la frecuencia de la señal recibida se puede clasificar de manera muy aproximada a una clasificación manual el índice S4, gracias al uso de técnicas de machine learning como árboles de decisión y support vector machine. Existen otros trabajos en los que se hace estadísticas de las cintilaciones (Statistics of GPS scintillations over South America at three levels of solar activity). Sin embargo el presente trabajo pretende hacer el estudio del último ciclo solar.

2. MARCO TEÓRICO

2.1. Cintilaciones

2.1.1. Definición

La cintilación o centelleo en la ionósfera es un cambio rápido en la fase y/o amplitud de una señal de radiofrecuencia a medida que pasa a través de irregularidades de densidad plasmática a pequeña escala en la ionósfera, como se puede observar en la Figura 2.

Las cintilaciones de las ondas de radiofrecuencia afectan a la potencia y la fase de la señal. El centelleo es causado por una estructura a pequeña escala (decenas de metros a decenas de km) en la densidad de electrones en la ionosfera a lo largo de la ruta de la señal y es el resultado de la interferencia de ondas refractadas y/o difractadas (dispersas). El centelleo generalmente se cuantifica mediante dos índices: S4 para el centelleo de amplitud y ϕ (sigma-phi) para el centelleo de fase. Los índices reflejan la variabilidad de la señal durante un periodo de tiempo, generalmente de un minuto.

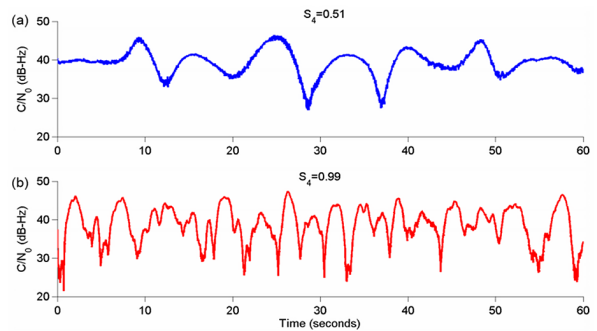


Figura 2. Dos ejemplos de medición de observaciones de intensidad de señal (C/N0) para diferentes medidas de S4. (a) S4=0.51 y (b) S4=0.99, se observa que un mayor valor de S4 representa una mayor variabilidad de la señal.

2.1.2. Cálculo de cintilación

La intensidad de los centilleos de amplitud generalmente se cuantifica mediante una métrica llamada índice S4. El índice S4 es la relación entre la desviación estándar de la potencia de la señal y la potencia de la señal media calculada durante un periodo de tiempo, que es:

$$S_4 = \sqrt{\frac{\langle I^2 \rangle - \langle I \rangle^2}{\langle I \rangle^2}}$$

Donde I representa la intensidad de la señal.

2.2. Cálculo de la posición

1. El cálculo de posición de un receptor GNSS se realiza basado un principio geométrico. Este principio consiste en la intersección de 4 esferas en el espacio donde el centro de estas esferas está determinado por la posición de los satélites GNSS. La intersección de esferas de forma ideal deberá determinar un solo punto (Figura 3.), sin embargo, de forma práctica la intersección de estas 4 esferas resulta en un espacio, para aumentar la precisión de este cálculo, es decir, reducir este espacio, se deben aplicar diversas correcciones. Antes de pasar al cálculo de la posición primero se describirán los tipos de mensaje y observables que envía un satélite GNSS, debido a que precisamente es de estos mensajes de donde se obtiene la información necesaria para el cálculo.

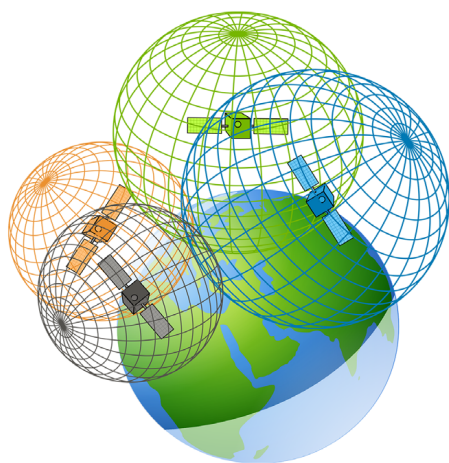


Figura 3. Explicación gráfica del principio usado por los GNSS para la estimación de la posición de los receptores.

3. RESULTADOS

3.1. Dataset

Para poder analizar los datos de errores en el cálculo de receptores GNSS y correlacionarlos con el parámetro S4 (cintilación de amplitud) y otras variables ionosféricas, en este caso, se usó un dataset con las variables mostradas en la Tabla 1 que indican actividad solar (f10.7), variaciones del campo magnético (BX, BY, BZ y Kp), condiciones del viento solar (SW), y nivel de radiación en función de día, hora y año.

	Variables	Formato
1	Año	I4
2	Día del año	I4
3	Hora	I3
4	Vector B Magnitud nT	F6.1
5	Bx nT (GSE, GSM)	F6.1
6	BY, nT (GSE)	F6.1
7	BZ, nT (GSE)	F6.1
8	SW Proton Density, N/cm ³	F6.1
9	SW Plasma Speed, km/s	F6.0
10	Índice Kp	I3
11	Índice f10.7	F6.1
12	Índice AE, nT	I5

Tabla 1. Direcciones de memoria virtual usadas en el programa

Además, se estableció como marco de referencia el último ciclo solar que va desde el 2009 hasta el 2019, y las estaciones de Huancayo y Jicamarca por ser las estaciones con mayor cantidad de continuidad en el registro de datos.

3.2. Gráficos

Se generaron gráficos polares, para poder observar una mayor variabilidad en el índice S4. Se configuró el gráfico de forma tal

que pueda mostrar escalas de colores para la escala de valores de índice S4, como observamos en la Figura 4, en esta figura se representa con color negro S4=0 y gradualmente se torna a rojo para el mayor valor de S4 analizado, que en este caso fue S4=1.4. Este gráfico nos permite analizar si existe alguna perturbación física, como por ejemplo una antena o un edificio cerca a la estación) o algún comportamiento repetitivo para ángulos y/o elevaciones determinadas. Por ejemplo, en la Figura 4 se observa un comportamiento extraño para un azimut entre 270° y 280° por ello estos datos no deberían ser tomados en cuenta para el análisis.

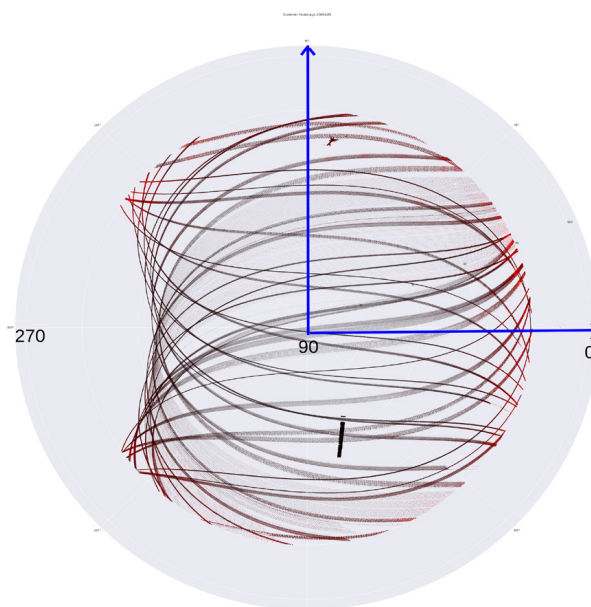


Figura 4. Gráfico polar que muestra ocurrencia en función al azimut y elevación, en color se muestra la intensidad del índice S4.

Además, se generaron histogramas como podemos ver en la Figura 5, que muestran el valor del índice S4 en función de la hora local, la cantidad de ocurrencias es representada por la barra de colores que se muestra, donde el color rojo denota más de 2000 ocurrencias y el color azul menos de 500 ocurrencias.

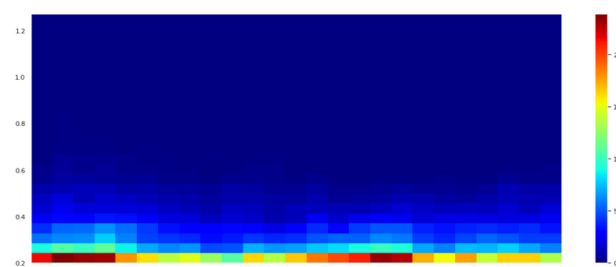


Figura 5. Gráfico polar que muestra ocurrencia en función al azimut y elevación, en color se muestra la intensidad del índice S4.

Además, luego de la obtención del nuevo dataset se realizó un nuevo gráfico que compara todas las variables, como por ejemplo S4, Kp, f10.7, AE entre otras, este tipo de gráficos es usado para determinar de forma parcial, tendencias entre variables y de esta forma ajustar los modelos de machine learning.

Además se generó una tabla que correlaciona variables utilizando el índice de Pearson.

4. DISCUSIÓN

El principal resultado que se busca es cuantificar errores en el cálculo de la posición GNSS y lograr correlacionar estos errores con índices de variables o fenómenos ionosféricos.

Hasta este punto se tienen todos los datos del último ciclo solar procesados, en este punto de la investigación corresponde aplicar diversos modelos de machine learning y evaluar la mejor técnica con diversos índices o funciones de penalización, se considera aplicar Support vector machines (SVM), árboles de decisión (decision tree) y random forest como primeros modelos, estos modelos serán aplicados por temporadas y dependiendo de su rendimiento se aplicará ensemble learning, además de ello se evaluará si existen patrones temporales, los métodos aún no están definidos, pero probablemente aplicar redes neuronales para este tipo de análisis sea óptimo.

Se observa por ahora que la variable con la mayor correlación con la variable S4 es la hora local y elevación, sin embargo, una vez aplicado un filtro a la elevación (eliminar bajas elevaciones), se observa que la mayor ocurrencia de cintilaciones se produce durante la noche y madrugada.

5. CONCLUSIONES

Los datos obtenidos representan el comportamiento normal mostrado por diversos trabajos, sin embargo, aún falta aumentar el dataset para correlacionar de una forma las variables, tal como se hizo en [1].

El cálculo de error en la posición representa un factor muy importante para esta investigación, observados los datos obtenidos mediante los archivos de posición (.pos) y los archivos (.S4), podemos armar que una tasa de muestreo de una posición por hora es insuficiente para correlacionar esta variable con el índice S4, dado que la medición de este índice se realiza en el orden de los minutos.

6. RECOMENDACIONES

Se recomienda utilizar para una primera aproximación modelos de árbol de decisiones (decision tree), support vector machines (SVM) y random forest.

Finalizada la implementación de estos modelos lo mejor es observar el rendimiento de cada modelo por periodo y aplicar un método de ensemble.

Para la elaboración de modelos obtenidos mediante técnicas de machine learning se recomienda explorar el uso de una cluster, debido a que con la capacidad computacional con la que actualmente se trabaja, una computadora i5, el procesamiento demora un día por estación.

7. AGRADECIMIENTOS

Al Instituto Geofísico del Perú por brindarme las facilidades para desarrollar mi tema de tesis.

Al Dr. Edgardo Pacheco por estar a cargo de mi capacitación en el Radio Observatorio de Jicamarca.

A todo el personal del Radio Observatorio de Jicamarca quienes han tenido mucho que ver con la construcción del sistema de observación.

8. BIBLIOGRAFÍA

[1] Ryan M. McGranaghan, Anthony J. Mannucci, Brian Wilson, Chris A Mattmann, and Richard Chadwick. New capabilities for prediction of highlatitude ionospheric scintillation: A novel approach with machine learning. *Space Weather*, 16(11):1817-1846, 2018.

[2] Linty, Nicola & Farasin, Alessandro & Favenza, Alfredo & Dosis, Fabio. (2018). Detection of GNSS Ionospheric Scintillations based on Machine Learning Decision Tree. *IEEE Transactions on Aerospace and Electronic Systems*. PP. 1-1. 10.1109/TAES.2018.2850385.

[3] Y. Jiao, J. J. Hall and Y. T. Morton, "Automatic Equatorial GPS Amplitude Scintillation Detection Using a Machine Learning Algorithm," in *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 1, pp. 405-418, Feb. 2017. doi: 10.1109/TAES.2017.2650758 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7812651&isnumber=7905775>

[4] Kung Chie Yeh and Chao-Han Liu, "Radio wave scintillations in the ionosphere," in *Proceedings of the IEEE*, vol. 70, no. 4, pp. 324-360, April 1982. doi: 10.1109/PROC.1982.12313 keywords: {Ionosphere; Geophysical measurements; Fluctuations; Frequency; Satellite broadcasting; Artificial satellites; Chaotic communication; Radar measurements; Phase measurement; Radar scattering}, URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1456581&isnumber=31312>

[5] Conker, R. S., El-Arini, M. B., Hegarty, C. J., and Hsiao, T., Modeling the effects of ionospheric scintillation on GPS/Satellite-Based Augmentation System availability, *Radio Sci.*, 38(1), 1001, doi:10.1029/2000RS002604, 2003.

[6] Space Weather Prediction Center [Online] <https://www.swpc.noaa.gov/phenomena/ionospheric-scintillation>

[7] J. Lee, Y. T. J. Morton, J. Lee, H. Moon and J. Seo, "Monitoring and Mitigation of Ionospheric Anomalies for GNSS-Based Safety Critical Systems: A review of up-to-date signal processing techniques," in *IEEE Signal Processing Magazine*, vol. 34, no. 5, pp. 96-110, Sept. 2017. doi: 10.1109/MSP.2017.2716406 keywords: {ionospheric electromagnetic wave propagation;safety;satellite navigation;ionospheric anomalies monitoring;ionospheric anomalies mitigation;GNSS based safety critical systems;signal processing techniques;global navigation satellite system;ionosphere induced error; safety critical navigation applications;Delays;Receivers;Decorrelation;Monitoring; ionosphere;Satellites}, URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8026593&isnumber=8026188>

[8] European GNSS Agency [Online] <https://www.gsa.europa.eu/european-gnss/what-gnss>

[9] De Oliveira Nascimento Brassarote, G., de Souza, E.M. & Monico, J.F.G. *GPS Solut* (2018) 22: 8. <https://doi.org/10.1007/s10291-017-0666-x>

[10] Li L, Zhang D, Zou Z et al. A dataset of ionospheric TEC and scintillation index from the Peking University station (2012). *Science Data Bank*. DOI: 10.11922/sciencedb.170.29

ESTUDIO DE LAS CAPAS E ESPORÁDICAS DE TIPO BLANKETING SOBRE EL RADIO OBSERVATORIO DE JICAMARCA

**Study of Blanketing Sporadic E layer layers over the Jicamarca Radio Observatory*

José Manuel Suclupe Osorio

Universidad Nacional de Ingeniería
Facultad de Ciencias
josesuclupe@uni.pe

Asesor: Dr. Edgardo Pacheco

Investigación Desarrollo e Innovación – Radio
Observatorio de Jicamarca - IGP

RESUMEN

Se ha reportado los resultados del primer estudio estadístico de la ocurrencia de las capas E esporádicas tipo Blanketing (*Esb*) sobre el Radio Observatorio de Jicamarca (ROJ) utilizando datos de la digisonda Digital Portable Sounder (DPS)-4 desde el 2001 al 2018. Se encontró que las capas *Esb* ocurren principalmente durante los meses de verano y también durante los años de mínima actividad solar. Se ha observado la ocurrencia principalmente entre las 0700 y 2000 LT con un pico máximo a las 1600LT y un segundo pico a las 0800-0900 LT. Se ha desarrollado un algoritmo y un software para identificar estos eventos *Esb* utilizando técnicas de procesamiento de imágenes y machine learning y se obtuvo una sensibilidad de 89%. Finalmente, se han usados datos de magnetómetros y el modelo de Muralikrishna y Kulkarni (2008) para discutir las posibles condiciones que favorecen la formación de estas capas *Esb*.

Palabras clave: E Esporádicas Blanketing, Ionogramas, Fuzzy C Means, Bosques aleatorios, Análisis de Componentes Principales, Modelo de Rotores, Procesamiento de Imágenes.

ABSTRACT

*This study presents the results of the first statistical study of blanketing sporadic E (*Esb*) occurrence over the Jicamarca Radio Observatory (JRO) using the Digital Portable Sounder (DPS)-4 digisonde data from 2001 to 2018. *Esb* occurs mainly during the summer months and also during the years of solar minimum. The occurrence of *Esb* is observed mainly between 0700 and 2000 LT with a maximum peak at 1600 LT and a second peak around 0800-0900 LT. An algorithm has been developed and software was implemented to identify *Esb* events using image processing techniques and machine learning algorithms getting a sensitivity of 89%. Finally, magnetometer data and the Muralikrishna and Kulkarni (2008) model have been used to discuss possible conditions that favor the formation of blanketing sporadic E.*

Keywords: *Blanketing Sporadic E, Ionograms, Fuzzy C Means, Random Forest, Principal Component Analysis, Rotor Model, Image Processing*

1. INTRODUCCIÓN

E Esporádica

De las distintas definiciones que existen para las capas **E** esporádicas (*Es*), vemos conveniente usar la definición de Resende et al. (2016) por ser general. Ellos las definen como regiones de incremento de la densidad de electrones, observadas entre 90 y 130 km de altura y constituidas de iones metálicos (como Mg^+ , Si^+ , Fe^+ , Ca^+ , Na^+), tienen diferentes mecanismos de formación de acuerdo a la región del globo donde son detectados (latitudes ecuatoriales, bajas/medias y auroral).

E Esporádica blanketing

Las capas *E* esporádicas son de tipo blanketing (*Esb*) cuando pueden bloquear parcial o totalmente las ondas de radio en las frecuencias transmitidas por las ionosondas para el sondeo de la ionósfera superior (entre 1.0 MHz y 14.0 MHz). La figura 1 muestra un esquema de las capas *E* esporádicas y su clasificación de blanketing y no blanketing. Se puede observar que en el caso blanketing ocurre una reflexión total por lo que a ciertas frecuencias no se puede sondear la región *F*.

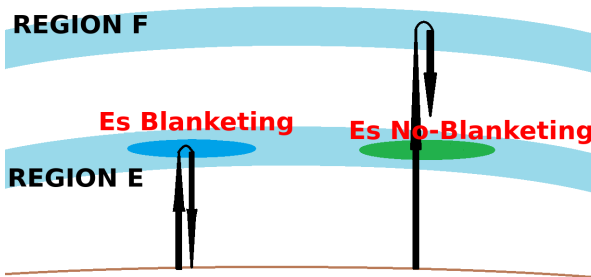


Figura 1: Capas *E* esporádicas blanketing y no blanketing.

La figura 2 muestra una serie de 5 ionogramas registrados por la digisonda DPS-4 cada 15 minutos el día 11 de enero del 2007 a partir de las 21:30:00 horas UTC. Cada ionograma es una matriz cuyo eje horizontal da la frecuencia y en el eje vertical valores de altura virtual, es decir es la distancia que recorrería la onda electromagnética en regresar como un eco asumiendo que la onda viaja a la velocidad de la luz en la ionósfera, el valor almacenado en cada posición de la matriz es la amplitud de la señal recibida almacenada en escala de colores.

En la figura 2 se observa cómo se presentan las capas *Esb* en los ionogramas. A las 21:30:00 horas no hay presencia de *Esb* y la capa *F* es observada entre 200 y 500 km, a las 21:45 horas se observa la repentina aparición de una capa *Esb* la cual oculta parcialmente la capa *F*, a las 22:00 permanece la capa *Esb*, a las 22:15 empieza a desvanecerse y a las 22:30 horas la capa *Esb* desaparece.

No existe una teoría que pueda explicar los mecanismos de formación de las capas *Esb*, en latitudes cercanas al ecuador magnético, como la región sobre el ROJ.

La mayoría de estudios estadísticos sobre las capas *E* esporádicas son de hace más de 40 años y abarcan eventos particulares o cortos periodos de tiempo, y la forma de identificación de las capas *Esb* en regiones ecuatoriales ha sido hecho de forma visual. No se ha realizado anteriormente un estudio estadístico de la ocurrencia de las capas *Esb* sobre el ROJ, probablemente debido a que el software disponible (SAO-XPLORER) no consigue calcular los parámetros ni identificar adecuadamente las capas *Es* debido a la presencia de ecos del Electrochorro Ecuatorial (EEJ) en la región *E* (80-150 km) durante el día, los cuales generan muchos falsos positivos. Por lo cual, se tuvo que implementar un nuevo software para la identificación automática de estos eventos *Esb*.

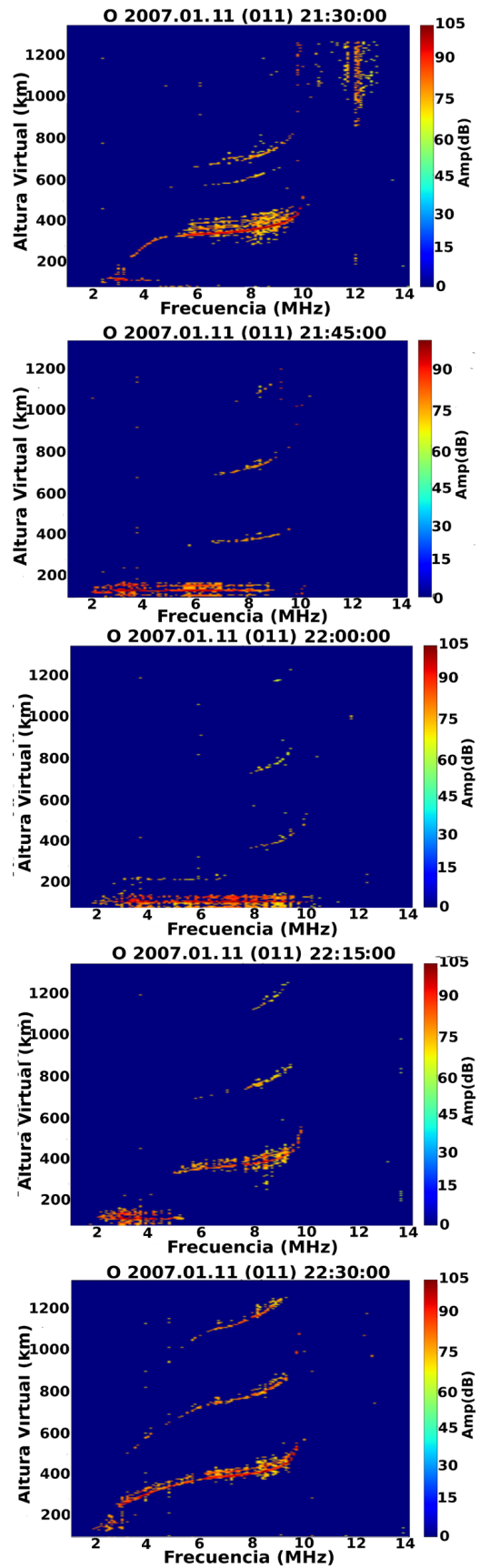


Figura 2: Serie de ionogramas registrados cada 15 minutos el 11/01/2007 a partir de las 21:30:00 horas UTC. A las 21:45, 22:00 y 22:15 se observa la presencia de *Es* blanketing.

Se usaron ionogramas de la digisonda DPS-4 del ROJ registrados cada 15 minutos desde el 2001 al 2018, lo cual permitió obtener conclusiones representativas. Se implementaron técnicas de

procesamiento de imágenes y machine learning para la identificación automática de estos eventos *Esb* clasificados como raros.

2. DESARROLLO

2.1. Algoritmo de identificación automática

El algoritmo de identificación de los eventos *Esb* se muestra en la figura 3. A continuación describimos los métodos usados y se pueden encontrar con más detalle en Suclupe (2019).

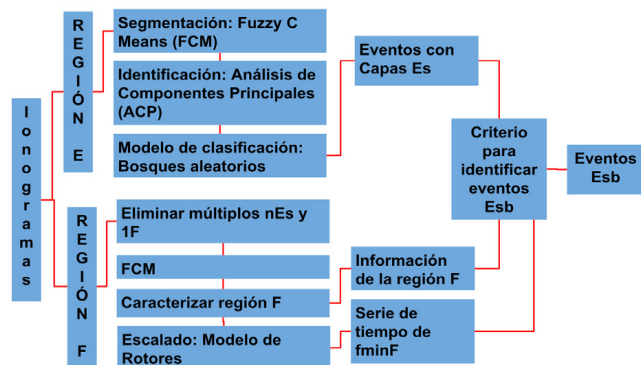


Figura 3: Algoritmo de identificación propuesto.

2.1.1. Región E

Fuzzy C Means es una técnica de clustering que permite agrupar elementos en grupos o clusters de acuerdo a un criterio de cercanías.

Esta técnica se aplicó a las regiones *E* de los ionogramas, con el fin de segmentar o separar los ecos debidos a las capas *Es* y los debidos al EEJ. La figura 4 muestra el resultado de aplicar el FCM. Debe notarse que el segundo cluster contiene los ecos de la *Es* y el resto de clusters contienen ecos debidos al EEJ.

Seguidamente se aplicó el método llamado análisis de componentes principales con el fin de obtener la pendiente de la capa *Es*, obtenido a partir del primer vector propio, y una medida del ancho de la capa, obtenida a partir del mínimo vector propio dados por este método.

Luego se aplicó el método de bosques aleatorios, con el fin de identificar las capas *Es* en la región *E* de los ionogramas. Este es un algoritmo supervisado de machine learning. Se usaron 10 variables explicatorias para el modelo, estas variables fueron la pendiente y el ancho de la capa *Esb* y otras 8 variables relacionadas con la morfología de la capa.

2.1.2. Región F

En esta región de los ionogramas, primero se realizó un filtro para eliminar los ecos que se presentan debido a las múltiples reflexiones de la onda electromagnética con la capa *Esb* y la Tierra.

Luego se aplicó otra vez el algoritmo FCM para segmentar los ecos de la región *F*. Esta segmentación permitió obtener información de la región *F* (es decir obtener si hay presencia de una traza de la capa *F* bien definida, si hay presencia de un fenómeno ionosférico llamado *F* dispersa o si no hay ecos obtenidos de la región *F*).

Después se aplicó una variación del modelo de rotores de Galkin (2004) con el fin de obtener una serie de tiempo de la frecuencia mínima de la capa *F*.

2.1.3. Criterio de identificación de las capas *Esb*

A partir de las tres salidas obtenidas con los algoritmos anteriores,

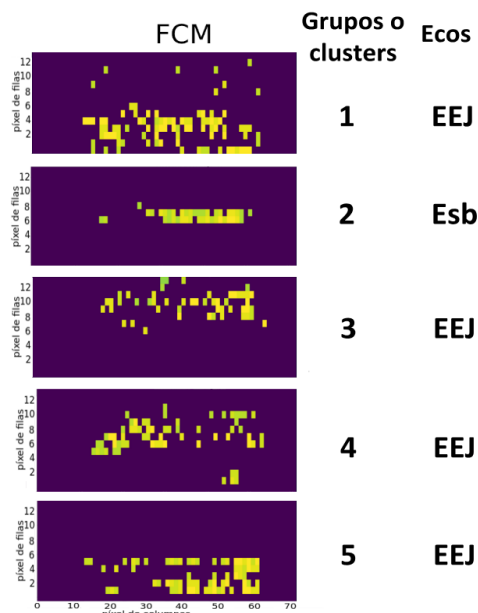
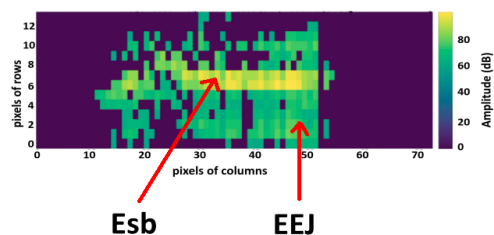


Figura 4: Fuzzy C Means aplicado a la región *E* de un ionograma registrado el 16/02/2008 a las 20:00 horas UT.

se propuso un criterio de identificación que tiene en cuenta la tendencia de la frecuencia mínima de la capa *F* (*fminF*) debido a la absorción de la región *D*. Este criterio tiene en cuenta los cambios bruscos de la *fminF* debido al bloqueo parcial o total de las capas *Esb*. Este cambio se mide a partir de la diferencia de la *fminF* con la curva parábola invertida ajustada en cada día, la cual representa el comportamiento de la absorción ionosférica. Entonces se tendrá una capa *Esb*, cuando el algoritmo de bosques aleatorios haya identificado una capa *Es*, y además esta diferencia sea mayor a un valor umbral igual a 0.5 MHz (para las capas *Esb* parcial) o cuando la información de la región *F* indique que no hay ecos en dicha región (*Esb* total).

El programa de identificación tiene una sensibilidad de 89%, una precisión de 72% y un F1-Score de 80%.

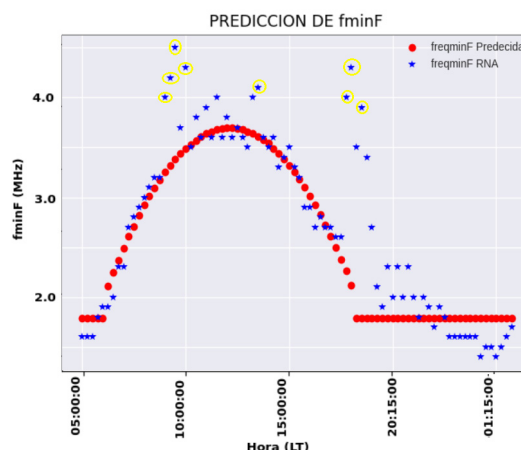


Figura 5: Serie de tiempo de la *fminF*. Se observa la tendencia de parábola invertida debido a la absorción de la región *D*.

3. RESULTADOS

De los eventos identificados automáticamente y luego de descartar los falsos positivos, se obtiene la siguiente estadística para la ocurrencia de las capas *Esb* en los años desde el 2001 al 2018.

La figura 6 muestra dos picos de ocurrencia, el primer pico a las 4 pm y el segundo pico alrededor de las 8-9 am.

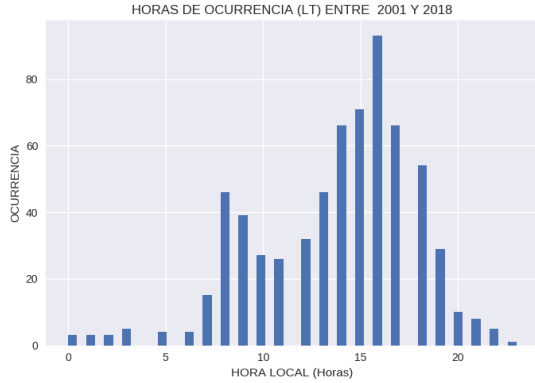


Figura 6

La figura 7 muestra que estos eventos se presentan principalmente en los meses del verano local.

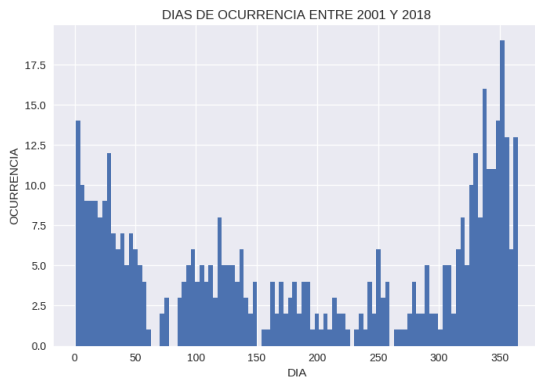


Figura 7: Histograma de días del año de ocurrencia.

La figura 8 muestra que los eventos *Esb* aparecen en un rango entre 90 y 135 km, con un pico máximo entre 95 y 100 km de altura.

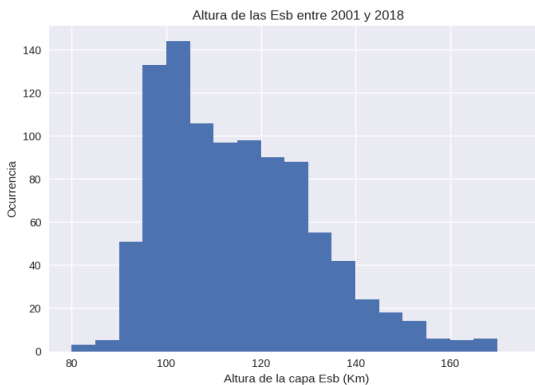


Figura 8: Histograma de alturas de ocurrencia.

La figura 9 muestra que estos eventos aparecen principalmente en los años de mínima actividad solar, es decir donde el sol presenta menor número de manchas solares.

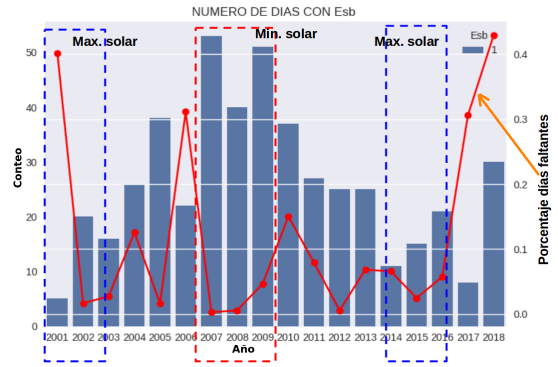


Figura 9: Histograma de años de ocurrencia. *Esb* ocurren más en años de mínima actividad solar.

Con el fin de medir si las capas *Esb* se presentaron en condiciones de EEJ o contraelectrochorro ecuatorial (CEJ) se utilizó la diferencia de la componente horizontal del campo magnético (ΔH) registrado en Piura (estación fuera del ecuador magnético) y el ROJ (estación en el ecuador magnético). Cuando el valor ΔH es positivo (negativo) se tiene el caso de EEJ (CEJ). A partir del cual se calculó el ΔH promedio en un rango de 6 minutos antes y después de la ocurrencia de un ionograma con *Esb*. Por otro lado se calculó la razón de este valor promedio entre el pico máximo (mínimo) si había condición de EEJ (CEJ).

La figura 10 muestra que estos eventos ocurren en principalmente en condiciones de EEJ débil (considerado ΔH menor a 25nT) y CEJ.

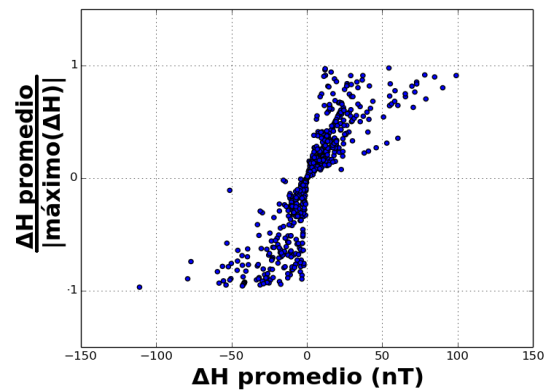


Figura 10: ΔH promedio (eje X) vs tasa de ΔH (eje Y).

4. DISCUSIÓN

A partir de los resultados obtenidos y del modelo desarrollado por Muralikrishna y Kulkarni (2008) quienes explican el efecto que tienen las partículas de polvo meteórico en la reducción de la intensidad del EEJ y la generación del CEJ, se ha propuesto tres condiciones que favorecen la formación de estas capas *Esb* en regiones ecuatoriales. En primer lugar la presencia adecuada de partículas de polvo meteórico, ya que mayor presencia de estos pueden capturar más electrones generando capas *Esb* más densas. En segundo lugar, un nivel adecuado de ionización, esto debido a que mayor ionización facilitará que más electrones sean capturados por los iones meteóricos. Esto último coincide con nuestras observaciones, debido a que en los meses de verano se tiene mayor ionización, así también en horas de la tarde se observa mayor ionización que en la mañana cuando recién empieza a incrementarse la ionización. En tercer lugar, se propuso una corriente de electrochorro ecuatorial normal no muy intensa. Esto es debido a que una corriente EEJ intensa,

generada por un campo eléctrico principal intenso, impulsaría a los electrones capturados por el polvo meteórico de tal forma que dejarían de estarlo.

5. CONCLUSIONES

Se ha obtenido la primera estadística de la ocurrencia de las capas Esb sobre el ROJ de datos registrados entre el 2001 y 2018. Se implementó un software para identificar automáticamente estos eventos raros. Se propusieron posibles condiciones que favorecen su formación.

6. AGRADECIMIENTOS

Al Instituto Geofísico del Perú por brindarme las facilidades para desarrollar mi tema de tesis.

Al Dr. Edgardo Pacheco por la asesoría durante este trabajo, al Dr. Marco Milla por las sugerencias dadas. Al Ing. Luis Condori por el préstamo de la GPU y junto al Ing. Fernando Villanueva y al Mag. Percy Córdor por su constante apoyo y sugerencias.

7. BIBLIOGRAFÍA

Galkin, I. A. (2004). A pre-attentive vision model for automated data exploration applied to space plasma remote sensing data (Doctoral dissertation, University of Massachusetts. Lowell).

Resende, L. C. A., Batista, I. S., Denardini, C. M., Carrasco, A. J., de Fátima Andrioli, V., Moro, J., ... & Chen, S. S. (2016). Competition between winds and electric fields in the formation of blanketing sporadic E layers at equatorial regions. *Earth, Planets and Space*, 68(1), 201.

Suclupe, J., Pacheco E., Córdor P, y Cárdenas D. (2019, junio). First statistical study of blanketing sporadic E over the Jicamarca Radio Observatory using image processing techniques and machine learning algorithms. Póster presentado en el Workshop CEDAR 2019, Nuevo México, EEUU.

Muralikrishna, P., & Kulkarni, V. H. (2008). Modeling the meteoric dust effect on the equatorial electrojet. *Advances in Space Research*, 42(1), 164-170

GENERADOR DE PULSOS SINCRONIZADOS CON GPS USANDO API REST

* *Generator of gps synchronized pulses using rest api*

Juan Marcos Luna Jaramillo

Pontificia Universidad Católica del Perú
Facultad de Ciencias e Ingeniería
jmlunajaramillo@gmail.com

Asesor: Joaquín Verástegui

Área de Investigación, Desarrollo e
Innovación (I&DI)
Radio Observatorio de Jicamarca - IGP

RESUMEN

En el presente trabajo se describe la realización de un generador de pulsos sincronizados con una señal PPS capaz de configurarse vía Ethernet. Este dispositivo se encargará de generar a la salida señales sincronizadas con la señal de entrada, la cual es una señal cuadrada con frecuencia de 1 Hz (PPS) proveniente de un equipo GPS. Se utilizará una tarjeta Tiva TM4C1294 que proporciona una señal de salida tipo PWM de 3.3V, y por medio de un Shield, el cual será descrito más adelante, esta será replicada y se obtendrán las salidas TTL requeridas, las cuales también serán señales cuadradas con periodo y ancho programables vía Ethernet. Y, por último, se describirá el desarrollo de una API REST que servirá para la configuración del dispositivo.

Palabras clave: [GPS] [Sincronización] [Periodo] [Duty] [API] [REST]

ABSTRACT

This work describes the realization of a pulse generator synchronized with a PPS signal capable of configuring via Ethernet. This device will generate output signals synchronized with the input signal, which is a square signal with frequency of 1 Hz (PPS) from a GPS. It will use a Tiva TM4C1294 card that provides a PWM output signal of 3.3 V, and by means of a Shield, which will be described later, this will be replicated and the required TTL outputs will be obtained, which will also be square signals with period and Wide programmable via Ethernet. And finally, the development of a REST API that will be used to configure the device will be described.

Keywords: [GPS] [Synchronization] [Period] [Duty] [API] [REST]

1. INTRODUCCIÓN

Los radares perfiladores de viento, como los que posee el Radio Observatorio de Jicamarca (ROJ) en distintas ubicaciones del Perú, son radares muy útiles para el estudio de fenómenos meteorológicos y como incide esto en el accionar de los vientos, el cual es un problema muy grande sobretodo en nuestro territorio, ya que poseemos un ecosistema que desarrolla diversos tipos de climas y es importante conocer cómo afecta esto a los vientos de nuestro país y sus características dependiendo del tipo de clima, para poder prevenir de alguna forma desastres.

El ROJ cuenta con diversos radares perfiladores de viento como es el caso del radar BLTR ubicado en el Radio Observatorio de Huancayo. Estos radares requieren un modo de activación basado en una secuencia de pulsos previamente programados, tal que pueda muestrear la información de los vientos estudiados en momentos definidos. Este tipo de señales se generan por un dispositivo llamado Generador de Pulsos sincronizadas por GPS, ya que dichas señales deben estar alineadas a la señal PPS del módulo GPS. En este trabajo, se procederá a explicar su diseño, construcción y programación.

Este dispositivo cuenta con una entrada, la cual es la señal cuadrada de 1 pulso por segundo (PPS) proveniente del GPS, y 4 salidas cuadradas con periodo y ancho programable que estarán sincronizadas a la señal de entrada. De dichas salidas, 2 de ellas serán enviadas en su estado natural (ondas TTL); mientras que las otras dos serán enviadas invertidas (luego de haber pasado por un dispositivo inversor). Asimismo, el periodo y ancho variable de dichas señales tiene una resolución temporal de 1us; es decir, se pueden realizar variaciones de ambos parámetros en múltiplos enteros de 1us. Estos parámetros serán configurados mediante un API REST.

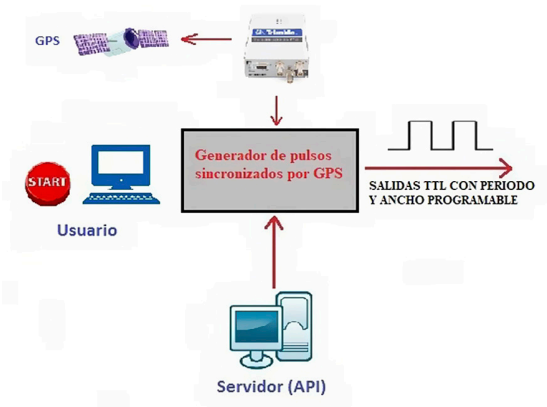


Figura 1: Esquema del dispositivo

2. DESARROLLO

2.1. Descripción del sistema

Las características que presentará el dispositivo que se desarrollará serán las siguientes:

De Hardware:

- 4 salidas TTL programables vía Ethernet: dos de las cuales serán las salidas positivas del generador (0V-5V-0V); mientras que las otras dos serán las salidas negadas del generador (5V-0V-5V).
- 1 entrada de referencia PPS con niveles TTL proveniente del equipo GPS
- Puerto Ethernet 10/100 para la conexión con el servidor API.

- Display LCD para la visualización de los parámetros de red y los datos del generador.

De Software:

Plataforma de configuración Ethernet mediante la creación de una API REST, por medio de la cual se podrán configurar parámetros del generador tales como:

- “Start” para el inicio del generador.
- “Stop” para la detención del generador.
- Periodo de las salidas
- Duty cycle de las salidas
- Número de ondas de la trama de salida
- Parámetros de red (IP, DNS, Gateway)

Cabe resaltar que por medio de esta API; además, se podrá obtener información actual de los parámetros del generador.

2.2. Herramientas utilizadas

En las siguientes líneas se conocerá más acerca del hardware y del software utilizados durante el proyecto, y que fueron herramientas muy útiles.

Hardware utilizado

Los instrumentos físicos utilizados para el desarrollo del proyecto se describen a continuación.

1. Tiva C TM4C1294

La tarjeta de evaluación LaunchPad conectado TM4C1294 de la serie Tiva™ C de Texas Instruments es una plataforma de evaluación de bajo costo para los microcontroladores basados en ARM Cortex-M4. Del diseño de LaunchPad conectado destaca el microcontrolador TM4C1294NCPDT con su MAC de Ethernet 10/100 en chip y PHY, USB 2.0, módulo de hibernación, modulación de ancho de pulso de control de movimiento y una multitud de conectividad simultánea en serie⁽¹⁾.

Las características de este microcontrolador son las siguientes:

- MCU TM4C1294NCPDT
- CPU ARM Cortex- M4 de 120MHz
- Flash de 1MB, SRAM de 256 KB y EEPROM de 6KB.
- MAC+PHY Ethernet 10/100 integrado.
- USB H/D/O y muchas interfaces adicionales de comunicación serial para UART, I2C, SPI.

2. Trimble Thunderbolt Kit

Es un equipo GPS que permitirá la sincronización de diversos dispositivos con respecto a alguna señal proveniente de este módulo. Sus características son las siguientes:

- Señal de reloj de 1Hz (PPS)
- Señal de reloj de 10MHz
- Conector RS232 Para la comunicación serial del equipo GPS con algún otro dispositivo.

Software utilizado

Para la realización del proyecto se utilizaron diversos programas, los cuales se describirán a continuación.

1. Energía IDE

Energía IDE es una plataforma de código abierto para el desarrollo electrónico, creado con la finalidad de ofrecer la misma estructura y conexiones con las que cuenta Arduino a las placas de desarrollo Texas Instruments MSP430 basados en LaunchPad. El IDE de Energía es multiplataforma, ya que puede usarse en Mac OS, Windows, y Linux. Usa el compilador 'mspgcc' que está basado en la estructura de Arduino.

2. Postman

Es una aplicación que permite enviar peticiones a servicios o APIs RestFul. Esta aplicación en particular permite realizar diferentes tipos de peticiones, tales como GET, POST, DELETE, PUT, sino que además se pueden lanzar varias de estas a la vez y no una por una, lo cual es algo que lo diferencia del resto de aplicaciones de este tipo. Se usará para realizar las pruebas de funcionamiento de los comandos API REST creados.

3. Eagle

Es un programa muy útil para el diseño de diagramas y PCBs. Con el fin de diseñar una placa, Eagle facilita dos ventanas, la primera es la ventana "esquemático", donde se creará el esquema de circuito que se usará; mientras que la segunda es la ventana "board", dentro del cual se podrá diseñar el PCB a partir del esquemático.

2.3. Desarrollo del Hardware

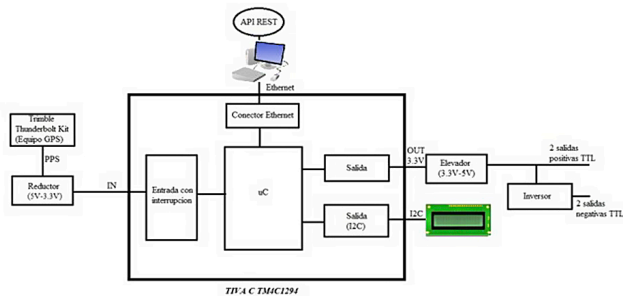


Figura 2: Diagrama de bloques del hardware

Para el desarrollo del proyecto se construyó y diseño el hardware, cuyo digrama de bloques se muestra en la Figura 2.

- El bloque "Reductor 5V-3.3V" sirve para reducir el valor de la salida TTL del GPS a 3.3V, el cual es el máximo soportado por el microcontrolador. Esta conversión se realizará haciendo uso de un circuito OR.
- El LCD con el que se trabaja es un NHD-C0220BiZ-FSW-FBW-3V3M, Character Lyquid Cristal Display Module. Este LCD se comunica por I2C, por lo que debe estar conectado a pines I2C del microcontrolador.
- Las salidas necesitan ser del tipo TTL (HIGH:5V); sin embargo, la salida del microcontrolador tienes un valor máximo de 3.3V, por lo que se requiere un "Elevador 3.3V-5V", Esta conversión se realizará con un circuito "Level Shifter" basado en transistores MMBT2222.
- Asimismo, se requerían dos salidas positivas y dos salidas negadas, por lo que para obtener las salidas negadas se utilizó un bloque Inversor, formado por una compuerta NAND del tipo NC7SZ00.

Adicionalmente, se diseñó un Shield para la tarjeta donde se colocará toda la circuitería requerida para el proyecto. En la Figura 3 se muestra el esquemático del Shield.

2.4. Desarrollo del Software

A continuación, se procederá a describir los códigos generados para el funcionamiento del programa.

2.4.1. Comandos para la API REST

Los comandos creados para nuestro propósito están basados en los métodos GET y POST, los cuales están definidos en formato JSON. Los comandos de tipo GET serán usados para solicitar información del estado del dispositivo. Por otro lado, los comandos del tipo POST se usarán para modificar valores de los parámetros usados por el generador, tales como: periodo, duty, numwaves, información del servidor. En la Tabla 1 se muestra una tabla con la definición de los comandos REST usados y su forma de uso.

Request	Type	Comments
status	Get	"Verifica la conexión entre el microcontrolador y la aplicación REST, y proporciona la información actual de los parámetros del generador."
changeip	Post	Configura nuevos parámetros de red.
reset	Post	Reinicia los valores de todos los parámetros a los valores definidos por defecto.
start	Post	Indica el inicio de la generacion de señales.
stop	Post	Indica que debe detenerse la generacion de las señales
period	Post	Configura el periodo de las señales PWM generadas
dutyCycle	Post	Configura el duty cycle de las señales PWM generadas
numwaves	Post	Indica el numero de ondas que deben generarse en la trama asignada

Tabla 1: Comandos REST

```
>>> import pgapi
>>> pg=pgapi.PGapi('10.10.50.200')
>>> pg.start()
{'u'Start': u'Ok'}
>>> pg.stop()
{'u'Stop': u'Ok'}
>>> pg.period('200')
{'u'Period': u'Ok'}
>>> pg.dutyCycle('20')
{'u'dutyCycle': u'Ok'}
>>> pg.numwaves('30')
{'u'numwaves': u'Ok'}
>>> pg.start()
{'u'Start': u'Ok'}
>>> pg.stop()
{'u'Stop': u'Ok'}
>>> pg.reset()
{'u'reset': u'Ok'}
>>> pg.start()
{'u'Start': u'Ok'}
```

Figura 3: Uso de comandos REST a través del terminal

Asimismo, se lograron realizar pruebas del funcionamiento del proyecto haciendo uso del código descrito en Python de la API REST, y se obtuvieron resultados esperados, como los mostrados en la Figura 3.

2.4.2. Código del programa

El programa está basado principalmente en el uso de Timers e interrupciones, siendo esta última muy importante para lograr priorizar la generación de las salidas requeridas sobre otras tareas.

Lo primero de lo que se encarga el programa es de configurar los Timers e interrupciones que se usarán; además, se configurarán los periféricos que serán utilizados, tales como el puerto por donde ingresará la señal PPS, el módulo I2C que usará el Display LCD, y el pin de donde se obtendrá la señal de salida. Asimismo, el programa cargará parámetros de red previamente almacenados en la memoria EEPROM, para su inicialización. Luego de realizadas las configuraciones, el programa verificará si existe algún cliente conectado al servidor, y si este cliente envió o solicitó algún dato por REST, donde dichos datos podrán actualizar los parámetros y funcionamiento del dispositivo.

Cuando se envía el comando "start", el cliente le indica al programa que debe habilitar la interrupción externa (la de mayor prioridad) para dar inicio a la generación de las salidas hasta el momento en que sea enviado el comando "stop", el cual detendrá la generación. El temporizador mostrará la información actual del dispositivo en su pantalla cada cierto intervalo de tiempo previamente definido.

Adicionalmente, se utilizan dos Timers los cuales se enfocan únicamente en darle forma a las señales de salida. Uno de ellos genera las señales PWM con periodo y ancho; mientras que el otro, es un contador que indica cuanto tiempo debe generar señales PWM y cuánto debe estar apagado, tal que se forme la trama requerida en la salida.

3. RESULTADOS

3.1. Shield en Eagle

En la siguiente figura se muestra la placa del Shield principal, la cual se mencionó en la sección de desarrollo y fue diseñada en Eagle (Circuito impreso).

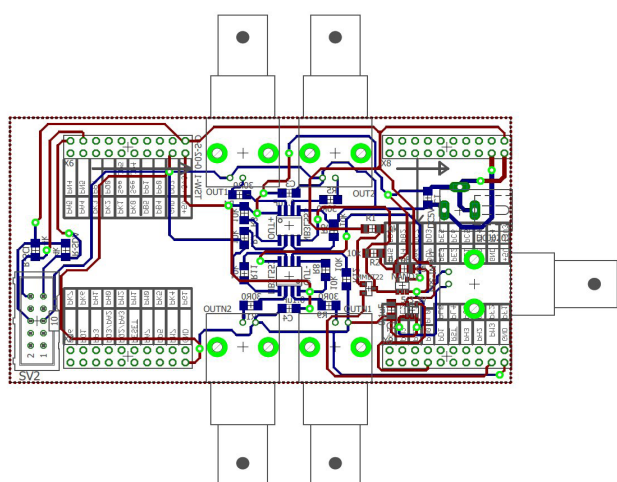


Figura 4: Placa del Shield

3.2. Salidas generadas

Se logró generar las señales cuadradas con periodo y ancho programable requeridas a la salida del dispositivo, sincronizadas con la señal de un PPS (un pulso por segundo). En la Figura 5 se muestra el funcionamiento del generador, siendo la señal ubicada en la parte posterior la señal PPS, mientras que la señal en la parte inferior corresponde a la salida del dispositivo.

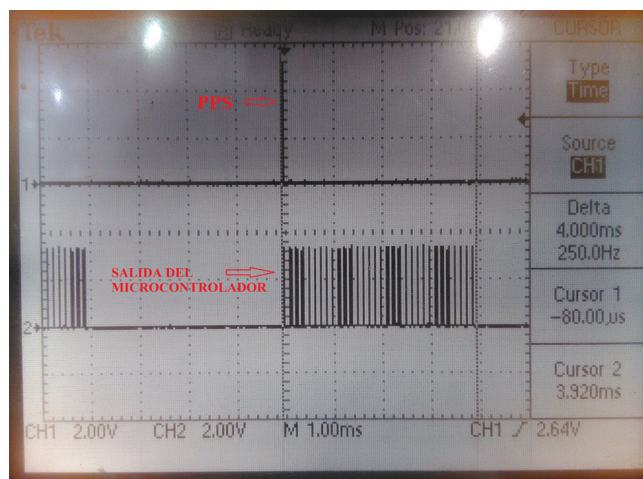


Figura 5: Señal generada con valores predeterminados (Periodo:125us, Duty:5us, NumWaves:32)

Asimismo, puede observarse la latencia que presenta el generador respecto a la señal PPS, siendo esta de 900ns con una variación entre 100 y 120ns, el cual es un valor admitido ya que es menor a 1us y es considerado un factor importante para el correcto desempeño del sistema. La Figura 6 muestra el desempeño del generador.

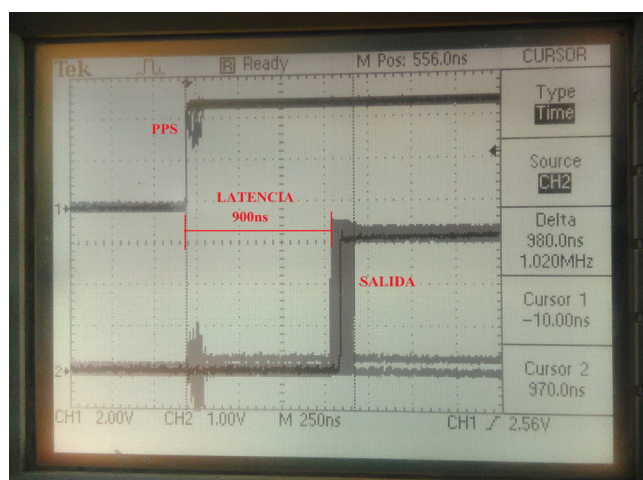
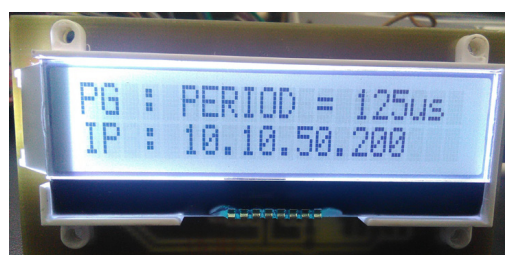


Figura 6: Desempeño de las salidas generadas

3.3. Funcionamiento del Display LCD

El periférico que se usó en este caso fue el Display LCD, el cual funcionó de manera correcta, mostrando los valores actuales de los parámetros del dispositivo, tales como: periodo (PERIOD), duty cycle (DUTY), número de ondas (NUMWAVES) y la información correspondiente a los parámetros de la red. En las figuras siguientes se muestran los 3 estados principales que mostrará la pantalla del LCD.



(a)

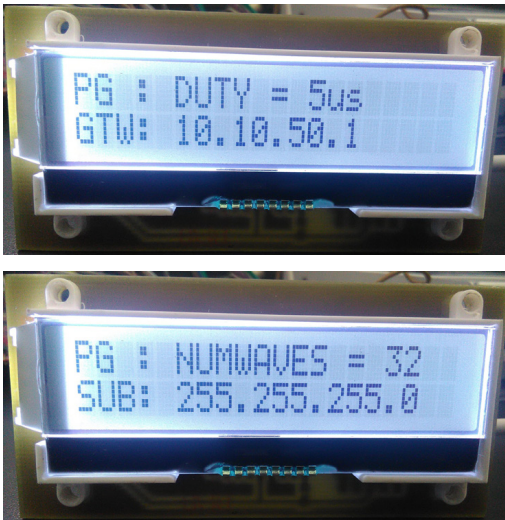


Figura 7: Estados principales del Display LCD

(b)

[2] Texas Instruments Incorporated (2013). TivaWare Peripheral Driver Library User Guide. Dallas, Texas.

[3] Texas Instruments Incorporated (2014). Tiva C Series TM4C1294 Connected Launchpad Evaluation Kit. Dallas, Texas.

[4] Trimble Navigation Limited Advanced Devices (2012). Trimble ThunderBolt E GPS Disciplined Clock User Guide. Sunnyvale, USA.

[5] Newhaven Display International (2014). NHD-C0220BiZ-FSW-FBW-3V3M COG (Chip-On-Glass) Character Liquid Crystal Display Module. Elgin IL, USA.

[6] Luis de La Cruz Painado (2017). Interfaz de Programación por Ethernet para Divisor de PPS. Radio Observatorio de Jicamarca, Lima, Perú.

(c)

[7] Texas Instruments Incorporated (2016). ARM® Cortex®-M4F-Based MCU TM4C1294 Connected LaunchPad™ Evaluation Kit. Dallas, Texas.

4. DISCUSIÓN

Para el proyecto, los compiladores del software Energía cumplieron una labor muy importante en la performance del dispositivo, ya que dependiendo del tipo del compilador usado durante las pruebas se observaban distintos resultados, unos más eficientes que otros.

Asimismo, el Display LCD presentó fallas a lo largo de las pruebas, generando diversos bugs en el desarrollo del programa, una solución para esto sería que el programa trabaje sin el LCD, lo cual evitaría problemas en el desempeño del generador.

5. CONCLUSIONES

Se consiguió generar las salidas sincronizadas con la señal PPS proveniente del GPS, obteniendo un tiempo de retardo de 900ns, con una variación de entre 100 y 200ns.

Se logró implementar el dispositivo tal que tenga una resolución temporal de 1us; es decir, se pueden realizar variaciones temporales con múltiplos enteros de 1us.

Se implementaron comandos REST, basados en los métodos GET y POST los cuales se comunicarán con el microcontrolador usando el puerto Ethernet.

6. RECOMENDACIONES

Cuando se deseen realizar cambios de los parámetros (period, dutycycle, numwaves) de las señales generadas, se recomienda detener la generación de los pulsos, usando el comando **'stop'** de la API REST, esto con el fin de que los cambios puedan procesarse y mostrarse de manera correcta a la salida. Luego de haber realizado los cambios deseados, se puede iniciar la generación de las salidas usando el comando **'start'** de la API REST.

7. AGRADECIMIENTOS

Mi agradecimiento al Instituto Geofísico del Perú a través del Radio Observatorio de Jicamarca por haberme permitido formar parte del Programa de Prácticas Pre-Profesionales, y un agradecimiento especial al Ing. Joaquín Verástegui por su apoyo constante y sus indicaciones, que permitieron que el proyecto alcance un mejor desempeño.

8. BIBLIOGRAFÍA

[1] Texas Instruments Incorporated (2015). Tiva TM4C1294NCPDT Microcontroller Data Sheet. Dallas, Texas.

GRABADO DE DATOS EN FORMATO DIGITAL RF SIGNAL PARA JARS 1.2

**Data recording in digital RF signal format for JARS 1.2*

Daniel Carrion Huacani

Universidad Nacional del Altiplano - Puno
Facultad de Ingeniería Mecánica Eléctrica,
Electrónica y Sistemas
daniconine@gmail.com

Asesor: John Rojas

Radio Observatorio de Jicamarca - IGP

RESUMEN

El presente trabajo hace una descripción del diseño e implementación de una aplicación que lee bloques de datos provenientes de un sistema de adquisición, éste es capaz de detectar la comunicación por protocolo TCP/IP por medio de un puerto Ethernet habilitado y la tarjeta NIDAQ (model.pcie-6537b). La aplicación procesa la data cruda (formato RAW) que proviene de JARS (Jicamarca Acquisition Radar System), para usarla en un formato estándar el cuál es hdf5, éste permite manejar una gran cantidad de información de una manera rápida, flexible y eficiente; todo esto utilizando un sistema jerarquizado de almacenamiento con Digital RF (Radio Frecuencia), el cual permite crear directorios y subdirectorios, donde se puede guardar ordenadamente y también permite la lectura de los datos hdf5.

Palabras clave: JARS, data cruda, hdf5, digital RF

ABSTRACT

The present work describes the design and implementation of an application that reads blocks of data coming from an acquisition system, this one is able to detect communication by TCP / IP protocol through an enabled Ethernet port and the NIDAQ card (model.pcie-6537b). The application processes the raw data (raw format) that comes from JARS (Jicamarca Acquisition Radar System), to use it in a standard format which is hdf5, it allows to handle a large amount of information in a fast, flexible and efficient way; all this using a hierarchical system of storage with Digital RF (frequency ratio), which allows to create directories and subdirectories, where it can be stored neatly and also allows the reading of the data hdf5

Keywords: JARS, raw data, hdf5, digital RF

1. INTRODUCCIÓN

El radar es un sistema de emisión y recepción de ondas de radio frecuencia que son emitidas al espacio atmosférico, y más exactamente la capa de la ionosfera. El Radio Observatorio de Jicamarca cuenta con varios sistemas de radares, entre las que destaca JARS, cuenta con una etapa de generación de señal RF, la siguiente etapa es de transmisión, en la que la señal se acondiciona y amplifica para ser transmitida por la antena a una potencia que llega hasta los 50 MW; la etapa de recepción, consiste en captar el eco de las ondas de radiofrecuencia por la misma antena, previo a esto se toma como referencia los parámetros (altura, tasa de muestreo, el IPP, el número de canales, la comprensión, etc), a las cuales fue generado la señal de radiofrecuencia emitida, gracias a estos parámetros es que se puede configurar el software y hardware de la última etapa del sistema radar, que es la adquisición.

Las señales de eco captadas por la antena son acondicionadas para su digitalización, y convertidos en data cruda (formato RAW) que poseen un peso en considerable en tamaño, y forman cadenas de valores consecutivos, donde no se puede diferenciar canales, la parte imaginaria o real de un dato; estos datos son estructurados en bloques.

Librería ZeroMQ

ZMQ es una librería de mensajería, posee un transporte eficiente de datos en TCP; que es lo que se usó para la transmisión de JARS a la Nidaq de la PC donde se encuentra el script de la Aplicación

Digital RF

Digital RF abarca un formato HDF5 estandarizado para leer y escribir datos de radiofrecuencia y el software para hacerlo. Puede realizar escritura y lectura de los datos de una forma aleatoria que facilita su búsqueda

Donde:

- "Chi RE" la parte real del canal "i"
- "Chi IM" la parte imaginaria del canal "i"
- "Hi" (#NSA) parámetro que indica el rango de la altura
- "Profile m" tiene un número de NSA
- "Block n" bloques de datos

Digital RF sobre Visual Studio

Al inicio del proyecto se procedió a compilar las librerías de Digital RF en Visual Studio 2017; para ello, se usó el compilador CMAKE GUI 3.4 y el propio compilador de Visual Studio (NINJA).

La utilización de Visual Studio se debió a la que JARS está montado en Visual Studio en lenguaje C; por lo que Digital RF en el lenguaje C, se iba agregar como una librería; los ejemplos de Digital RF, no compilaban adecuadamente arrojando al final de la ejecución con los ".dll".

La ilustración 2, muestra la ejecución final de digital.dll, en los scripts modificados en Visual Studio 2017; para guardar archivos hdf5 con Digital RF.

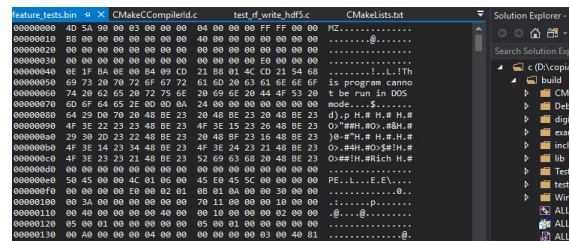


Ilustración 2. Datos binarios de la carpeta /Debug en build

2. DESARROLLO

2.1. Diseño de la Aplicación

Estructura de los datos de JARS

El software de adquisición JARS, está encargado de configurar los parámetros del experimento a realizar con el radar, y adquirir las señales digitalizadas por medio de la tarjeta de Adquisición de Datos de National Instrument (Nidaq) y guardar la información generada. El modo para ejecutar el software es en modo batch y con una consola se interactúa con el usuario.

Para configurar el sistema, al inicio de cada experimento, se habilita los canales y la programación de los receptores que se va utilizar; esto se realiza leyendo el archivo ".racp" lo cual contiene los datos del experimento. Los bloques de datos de JARS poseen un encabezado (Header) y el segundo Bloque de Dato RAW que se muestra en la ilustración 1.

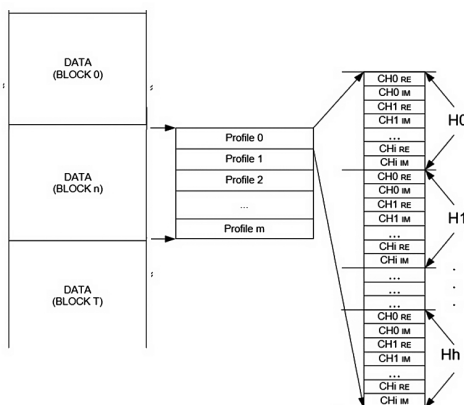


Ilustración 1. Estructura de los bloques de radar

Digital RF sobre Python

Ante los errores de compilación y ejecución de Digital RF en Visual Studio y en la consola de Windows (cmd.exe); se optó por realizar en el lenguaje de programación de Python, que se ejecutó adecuadamente con los ejemplos que viene en Digital RF.

En la ilustración 3, se puede apreciar la construcción de ejemplos y test en Digital RF, otro de los aspectos es que se vio mayor eficiencia en el uso de la consola de Linux (Ubuntu), para hacer mas eficiente la ejecución de la aplicación.

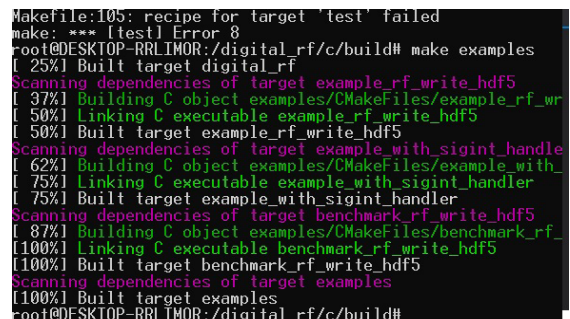


Ilustración 3. Compilación de los ejemplos Digital RF en Ubuntu

Windows y Python, los ejemplos del repositorio de digital RF, funciona adecuadamente; sin embargo cuando se realizó la simulación, se encontraron errores de "Cython" y error returned (0xC0000374); que son de sobrecarga en el buffer.



Ilustración 4. Software de Jars y los módulos que lo conforman

JARS sobre Windows

- El manual de instalación detalla la serie de pasos para la instalación del software de JARS.
- La instalación de los drivers de la placa NIDAQ se encuentra <www.ni.com/es-cr/support/model.pcie-6537b.html>

Una vez instalados todos los requerimientos para Digital RF, se procedió analizar las funciones y métodos de Digital_rf.

Para la simulación se hace modificaciones en el software de JARS, específicamente en el archivo JicaHeader.cpp, utilizando el Visual Studio 2010; se comento la línea 310 que definía el tamaño del buffer de la cabecera, línea 319 que realiza la operación de medir la cantidad de filas de datos y la línea 327 hace uso de ZmqSocket para enviar el tamaño del bloque con los datos.

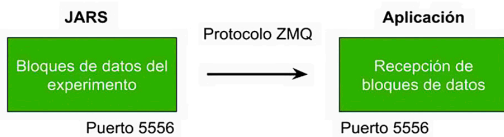


Ilustración 5. Diagrama de bloques de comunicación

Concluido la etapa de análisis; se procedió a diseñar el script que pueda guardar los datos hdf5 con los datos generados por JARS. Para lo cual se hizo uso de la opción simulación del software de JARS, ejecutando un archivo que contiene los datos de un experimento pasado, esto para generar los bloques de datos como si fuera la etapa de adquisición del radar.

En la ilustración 3, se representa las principales partes para la comunicación con la PC que hará el pre-procesamiento y guardado de datos.

Diagrama de Flujo

El script cuenta con una estructura orientada al tiempo; esto quiere decir que la hora obtenida por la programación y en la aplicación, deben de crear los archivos hdf5 con un subdirectorio que indique la fecha yy/mm/dd y la hora hh/mm/ss, esto es primordial para crear la estructura jerarquizada.

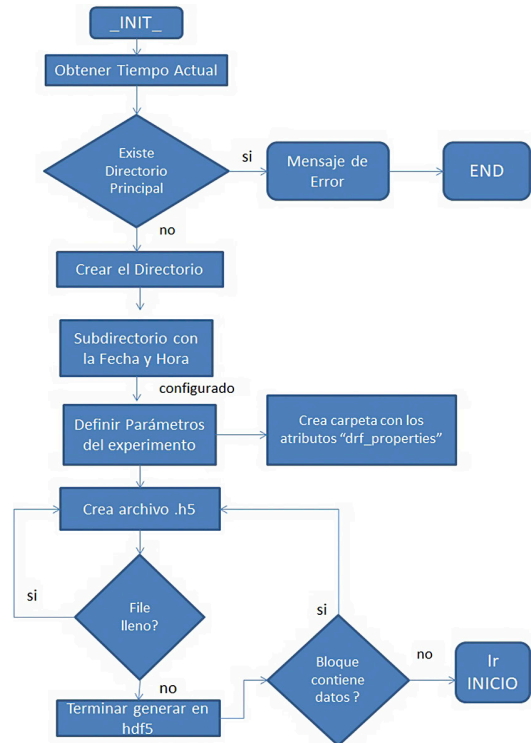


Ilustración 6. Diagrama de flujo de la Aplicación

La secuencia de la adquisición se da con el libreria ZMQ, la aplicación se implementa como se representa en la ilustración 7, de manera que conjuntamente con la anterior diagrama forman un bucle, esto con la finalidad de esperar la comunicación por el Puerto Ethernet, que por configuración debe ser el mismo en la salida de JARS y la PC de procesamiento.



Ilustración 7. Diagrama de Flujo de comunicación con ZMQ

2.2. Pruebas

El desarrollo de la aplicación llevo a realizar modificaciones leves, en como Digital RF adquiere el tiempo, por lo general JARS toma

como referencia a la hora de Zona Horaria, que esta adelantada en aproximadamente 5 horas.

En las pruebas el software JARS, envía los datos de un experimento realizado con una frecuencia de reloj 60 MHz esto a modo de simulación. Ejecutando el archive de Windows Batch File, que previamente se ha modificado con un editor de notas, cambiando el modo de operación del batch.

```

jrojas@CI-48: /media/ROJ/JARS1_1
ultimo directorio generado = /tmp/data/cho/2020-01-21T08-00-00/
UTC ultima marca de tiempo escrita es 1579596613
2020-01-21 03:50:13.440647
413940
413940
-----
Numero de Canales Configurado = 5 - Numero de muestras =20697
[[ ( 244, 0) (1103, 172) ( 0, -17590) ... ( 489, 489)
( 499, 499) ( 503, 503)]]
ultimo archivo .h5 escrito = /tmp/data/cho/2020-01-21T08-00-00/rf@1579596292.400.h5
ultimo directorio generado = /tmp/data/cho/2020-01-21T08-00-00/
UTC ultima marca de tiempo escrita es 1579596618
2020-01-21 03:50:18.116940
413940
413940
-----
Numero de Canales Configurado = 5 - Numero de muestras =20697
[[ ( 244, 0) (1103, 174) ( 0, -17585) ... ( 495, 495)
( 515, 515) ( 498, 499)]]
ultimo archivo .h5 escrito = /tmp/data/cho/2020-01-21T08-00-00/rf@1579596292.500.h5
ultimo directorio generado = /tmp/data/cho/2020-01-21T08-00-00/
UTC ultima marca de tiempo escrita es 1579596622
2020-01-21 03:50:22.748950
413940
413940
-----
Numero de Canales Configurado = 5 - Numero de muestras =20697
[[ ( 244, 0) (1103, 176) ( 0, -17581) ... ( 525, 525)
( 498, 498) ( 510, 510)]]
ultimo archivo .h5 escrito = /tmp/data/cho/2020-01-21T08-00-00/rf@1579596292.700.h5
ultimo directorio generado = /tmp/data/cho/2020-01-21T08-00-00/
UTC ultima marca de tiempo escrita es 1579596627
2020-01-21 03:50:27.356647
    
```

Ilustración 8. Pantalla de la consola ejecutando la aplicación

En la ilustración 8, se ve claramente el bloque de datos obtenido en hexadecimales, enviados mediante el protocolo ZMQ; la aplicación en este caso muestra la serie de datos, que aparentemente no cuenta, con un orden o señas, para su identificación.

Luego de recibido, la aplicación realiza un procesamiento, de acuerdo a los parámetros configurado en el archivo ". racp", que indica la secuencia del algoritmo para el número de canales, el tipo de dato.

En la consola se ve como se crea el ultimo archivo "file" de formato hdf5 con los parámetros que se configuro; también se muestra el ultimo subdirectorio creado y la última marca "UTC" que mide la hora actual, para crear el siguiente file.

3. RESULTADOS

Luego de realizar las simulaciones y la prueba en tiempo real con el módulo de JARS. Se genera los archivos hdf5; sin ningún inconveniente, la creación de los files en subdirectorios que contienen la fecha y hora exactas se da de manera ordenada; en la simulación se debe hacer mención que la máxima cantidad datos que puede manejar esta entre los 10 bits.

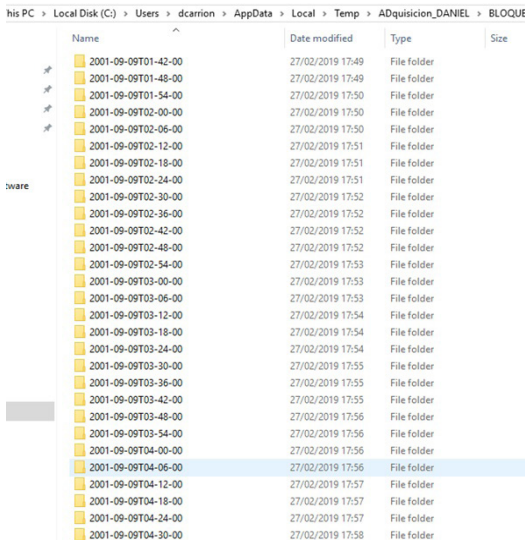
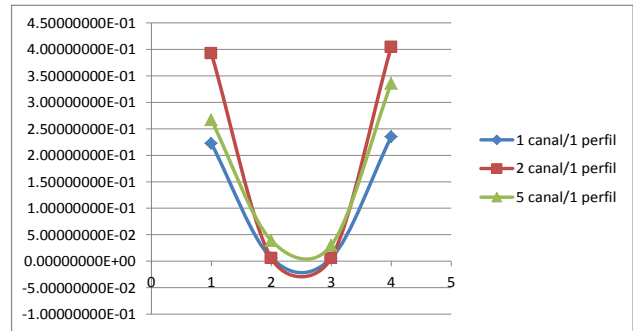


Ilustración 9. Subdirectorios y archivos hdf5

	Adquisición (ms)	Procesa (ms)	Escritura (ms)	Tiempo Total (ms)
1 canal/1 perfil	332.103825	0.2749	2.110887	334.489584
2 canal/1 perfil	332.6932	0.342274	3.562093	336.5976
5 canal/1 perfil	395.34683	0.626493	5.225492	401.198816

Table 1. Tiempos obtenidos en la ejecución del script a 200 NSA



El script tiene el mismo tiempo de adquisición con dos y un canal y hay muy poca diferencia con el de 5 canales.

4. DISCUSIÓN

La necesidad de poder ejecutar Digital RF en Windows, se debe a la necesidad de crear la menor cantidad de conflicto con JARS; en el desarrollo del proyecto al usar ZMQ se comprobó que algunas de las simulaciones botaban errores de tipo "Cython" que ocurrían con loa lectura de datos enormes de más 1,2 millones de bytes; lo que causaba el conflicto de las librerías de ZMQ en C y en Python.

De igual forma la lectura de las cabeceras para la generación de la metadata; necesita un procesamiento adicional; por lo que los recursos de lectura y escritura se ven afectados a parte del tamaño de los datos crudos que se procesa.

5. CONCLUSIONES

El tiempo de simulación recopilado en la ejecución del script muestra que la aplicación puede correr sin problemas para generar los archivos hdf5 guardando los datos provenientes de JARS.

El directorio principal, contiene los subdirectorios con la fecha y hora de las pruebas de simulación y reales de forma correcta

Digital RF, se acondiciona mas fácilmente a Linux que a Windows; Python corre mejor que con un entorno en lenguaje C (Visual Studio 2017); no logra correctamente ejecutar los test y ejemplos del repositorio de Digital RF

Los archivos generados son de solo datos, para generar la metadata se hace de forma estática con el archivo ". racp"

6. RECOMENDACIONES

Si bien JARS esta construido sobre una plataforma de Visual Studio para sistema operativo de Windows; debería migrarse a Linux, que daría una buena opción de desarrollo

Contar con información adicional a la generación de los formatos RAW, así como una actualización en el manual, con respecto a JARS 1.2

Estudiar el proyecto Digital_RF, esto es necesario para conocer los objetos que generen la secuencia jerarquizada de los datos y hallar la forma de que se pueda modificar el archivo drf_ properties que ayudaría en la elaboración de la metadata

7. AGRADECIMIENTOS

Al Instituto Geofísico del Perú por brindarme las facilidades para colaborar en un proyecto de investigación.

Al personal que labora en el Radio Observatorio de Jicamarca, y en especial al ingeniero John Rojas por la paciencia y la guía a lo largo de mis prácticas

8. BIBLIOGRAFÍA

ROBERTO GARRIDO. (12 Mayo 2011). Configurando proyectos multiplataforma fácilmente con CMake. 13 Mayo 2011, 02:55, de Genbeta Sitio web: <https://www.genbeta.com/desarrollo/configurando-proyectos-multiplataforma-facilmente-con-cmake>

Radio Observatorio Jicamarca (Junio 2010). SISTEMA DE ADQUISICIÓN DIGITAL- Recuperado - <http://jro-dev.igp.gob.pe:3000/projects/jars-1.0> JARS.

Ryan Volz. (13 de Junio 2018). Signal processing, radar – Digital RF, de GitHub.Inc [US]: https://github.com/MITHaystack/digital_rf

IMPLEMENTACIÓN DE RUTINAS PARA LA ESTIMACIÓN DE VIENTOS Y PRECIPITACIÓN DE UN RADAR PERFILADOR DE VIENTOS DE ANTENA ESPACIADA QUE OPERA EN UHF

**Implementation of routines for wind and precipitation measurements of a spaced antenna wind profiler radar that operates in UHF band*

Erick Bocanegra Salas

Universidad Peruana de Ciencias Aplicadas
Facultad de Electrónica y Telecomunicaciones
erick.bocanegra@jro.igp.gob.pe

Asesor: Dr. Danny Scipion

Investigador Científico: Radio Observatorio de Jicamarca - IGP

RESUMEN

El Instituto Geofísico del Perú, a través de su sede científica el Radio Observatorio de Jicamarca (ROJ), ha desarrollado un radar perfilador de viento llamado CLAIRE por sus siglas en inglés CLeAr-Air and Rainfall Estimation que opera en la banda UHF a 445MHz. El objetivo de este radar es de estudiar los fenómenos troposféricos y a través de procesamiento de señales determinar la velocidad y dirección de los vientos y cuantificar la cantidad de precipitación. Este trabajo describe los pasos para lograr las estimaciones deseadas. Los algoritmos están desarrollados en lenguaje Python y utiliza Signal Chain, biblioteca de programas que es utilizada para procesar los datos de los distintos radares del ROJ. Finalmente, los resultados deberán ser validados con los radares BLTR para vientos y MIRA-35C para precipitación.

Palabras clave: CLAIRE, Signal Chain, Perfilador de vientos, Antena Espaciada

ABSTRACT

The Geophysical Institute of Peru, through its scientific facility the Jicamarca Radio Observatory (JRO), has developed a wind profiler CLAIRE (CLeAr-Air and Rainfall Estimation) that operates in the UHF band at 445MHz. The objective of this radar is to study the troposphere and through signal processing determine the velocity and direction of winds and the amount of precipitation. The present work describes the steps followed to achieve the final estimates. The algorithms are developed in Python and uses Signal Chain (JRO open source) library that is used to process the data from the different JRO radars. Finally, the results will be validated with BLTR radar for wind and MIRA-35C for precipitation.

Keywords: CLAIRE, Signal Chain, Wind Profiler, Spaced Antenna

1. INTRODUCCIÓN

El radar CLAIRE por sus siglas en inglés CLeAr and Rainfall Estimation ha sido llevado al Observatorio de Huancayo para que pueda realizar mediciones de viento y precipitación. Este radar fue desarrollado como la posible solución ante la carencia de sistemas que permitan estudiar las condiciones climáticas instantáneas con alta resolución espacial y temporal para los investigadores del Instituto Geofísico del Perú (IGP).

Para el desarrollo de este radar, se dispuso la conformación de distintos grupos. Las funciones de cada grupo fueron las siguientes: el diseño del radar, la construcción del radar, la implementación, la interconexión entre el radar y los demás dispositivos y el procesamiento de los datos adquiridos por el radar. El desarrollo de esta tesis está enfocado en el procesamiento de datos de un radar que opera en la banda de UHF (Ultra High Frequency), con el diseño de antena espaciada, para la estimación de vientos y precipitación que son observados en un perfil vertical con diferentes rangos de alturas. Para este procesamiento se debió analizar cuáles eran las técnicas disponibles para la estimación de vientos y precipitación, y determinar la más óptima para el radar con las características mencionadas. Además, hacer una validación de los resultados para su validación.

Para este proyecto se hizo un estudio de la técnica Full Spectral Analysis (FSA) para la estimación de vientos que deriva de la técnica Full Correlation Analysis (FCA). También de la estimación del factor de reflectividad Z y la razón de lluvia R para hidrometeoros en la región de Rayleigh.

1.1. Objetivo

El objetivo principal de la tesis es poder desarrollar los algoritmos de procesamiento de datos para la estimación de la velocidad de los vientos en sus tres componentes (zonal, meridional y vertical) y el factor de reflectividad y razón de lluvia de un radar perfilador de vientos de antena espaciada que opera en UHF.

2. DESARROLLO

2.1. Radar CLAIRE

El radar CLAIRE es un radar perfilador de viento del tipo antena espaciada que opera a una frecuencia de 445 MHz. Su arquitectura consiste en cuatro arreglos de antenas cuasi hexagonales. Tres usadas como antenas receptoras y uno como transmisor. Su diseño fue planificado para ser un radar de bajo costo y fácil de replicar. En la figura 1, se presenta la arquitectura del radar CLAIRE. Éste presenta una separación entre sus antenas receptoras; la distancia entre ellas dependerá de la comparación entre el cruce espectral que se realizará con sus señales en su procesamiento.

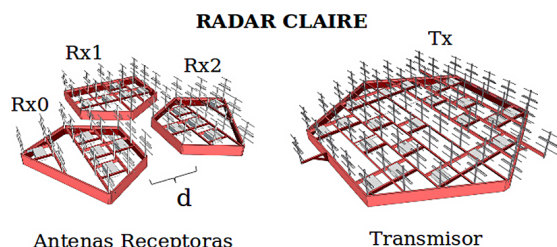


Figura 1. Arquitectura del Radar CLAIRE

Puesto que fue diseñado para funcionar en la banda UHF, este radar es sensible a los ecos de viento y precipitación. Los ecos de precipitación son visibles porque su longitud de onda λ se encuentra dentro del rango de aproximación Rayleigh.

2.2. Rutinas de estimación

La solución propuesta se presenta en el diagrama general de la Figura 2. Y las técnicas que se van a utilizar para la estimación de vientos (FSA) y precipitación (Z) de los ecos recibidos respectivamente.

Los datos obtenidos del radar deben presentar los ecos de viento y en eventos climáticos de lluvia. En ocasiones específicas se puede observar dos ecos con diferentes desplazamientos de frecuencia. Para poder estimar los vientos y la precipitación se debe procesar cada eco independientemente. Es por este motivo, que se debe agregar una unidad de procesamiento que discrimine estos ecos; para este caso, se consideró utilizar un algoritmo de selección de ecos y utilizar las señales obtenidas de cada uno de estos ecos para su procesamiento individual.

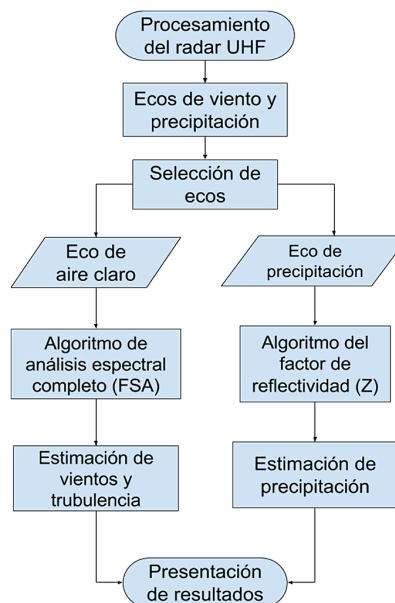


Figura 2. Diagrama de flujo general del procesamiento del radar UHF

2.3. Descripción del Software

La solución propuesta en el procesamiento de los datos obtenidos del radar presenta, en términos generales, cuatro etapas. La primera llamada etapa de configuración, que se encarga de ordenar a los bloques de datos obtenidos para su procesamiento. La segunda etapa es llamada la etapa de selección, donde se presentan los bloques de datos del radar que contienen información de interés y se seleccionan los datos que han sido detectados dependiendo de sus características espectrales. La tercera etapa es llamada estimación de parámetros, que se encarga de transformar los datos seleccionados en estimaciones de vientos o precipitación. Este procesamiento se realiza individualmente con datos previamente seleccionados. Por último, en la cuarta etapa, las estimaciones obtenidas con valores cuantitativos deberán ser mostradas de manera gráfica para su mejor comprensión. Estos datos deberán presentarse en línea y guardarse en una base de datos. La Figura 3, presenta el esquema detallado del proyecto a desarrollar. Además, esta etapa corresponde a la parte final del proyecto del radar UHF ya que, para esta instancia, el desarrollo del hardware y las pruebas de los datos obtenidos del radar ya fueron finalizadas.

En la Figura 3 se detallan cada etapa dentro de un recuadro. La configuración para generar los archivos .pdata que contienen los datos del radar en el dominio de la frecuencia. Datos que serán utilizados para estimar vientos y precipitación. Para distinguirlos se consideró que los vientos verticales presentan la característica de ser menores a 3 m/s, mientras que la velocidad de caída de los

hidrometeoros normalmente superan los 1.5 m/s. La estimación es independiente para cada caso y los resultados finales son subidos para ser mostrados en la base de datos de MADRIGAL.

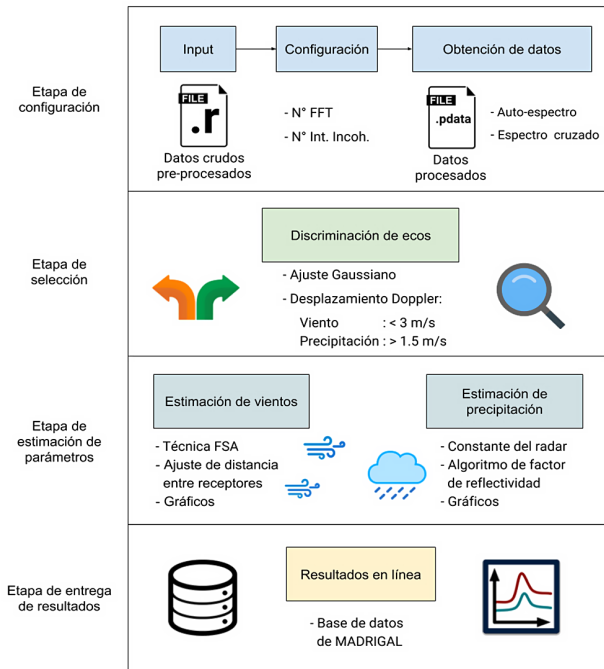


Figura 3. Esquema detallado del proyecto a desarrollar

2.4. Algoritmos desarrollados

2.4.1. Algoritmo de selección

Una vez obtenidos los auto-espectros y los espectros cruzados, se da inicio con el procesamiento de los algoritmos para la estimación de vientos y precipitación. Estos fenómenos atmosféricos presentan ciertas características que permiten identificarlos con un radar Doppler. El efecto Doppler indica que cuando un cuerpo está en movimiento produce un desplazamiento de su frecuencia que depende si éste se aleja o se aproxima al receptor. En el caso del radar UHF, la turbulencia que es sensible presenta un ligero desplazamiento vertical, producido por la evaporación de partículas que están en el aire o por el descenso de aire frío empujados por la evaporación. Estos desplazamientos del aire se encuentran normalmente entre -3 y 3 m/s de velocidad vertical, mientras que la precipitación presenta un desplazamiento Doppler mayor. Además, los hidrometeoros siempre van a presentar un desplazamiento Doppler positivo debido a la fuerza de gravedad. Puesto que, al ser partículas de mayor tamaño que las de turbulencia, su resistencia al aire también es mayor. Las velocidades de caída de los hidrometeoros pueden encontrarse desde los 1.5 a 15 m/s de velocidad de caída.

El algoritmo de selección de ecos, utiliza el bloque de datos como está cuando se trata de estimación de vientos, pero en casos de lluvias (que son menos frecuentes y por temporadas) se realiza un recorte de datos entre 0 y 1.5 m/s para evitar ecos de vientos ligeros y solo procesar los ecos de precipitación. Además, hace un corrimiento del bloque que sólo procese velocidades positivas. La figura 4 muestra un ejemplo de cómo se vería el bloque de datos con la señal de precipitación ya seleccionada para que sea estimada con el algoritmo de factor de reflectividad.

Para la estimación de vientos, se deja los bloques como están inicialmente y solo se agregan condiciones en el algoritmo FSA.

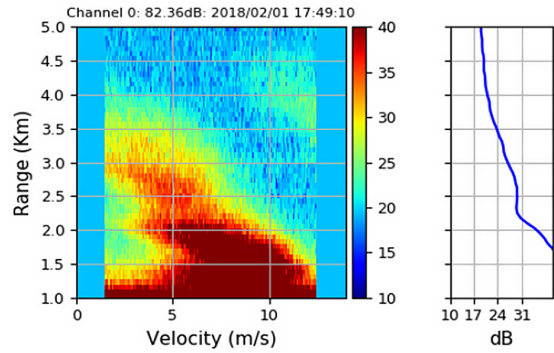


Figura 4. Ejemplo de selección de eco de precipitación

2.4.2. Algoritmo FSA

El procedimiento para la estimación de vientos a partir de datos de un radar de antena espaciada que opera en la banda UHF se muestra en la Figura 5 con la técnica FSA. La obtención de las velocidades con esta fórmula se obtiene de una ecuación general de la elipsoide presentada por Briggs.

Se deberá manipular los auto-espectros y espectros cruzados, además de conocer la distancia entre receptores más adecuada para obtener las velocidades horizontales y verticales de los vientos.

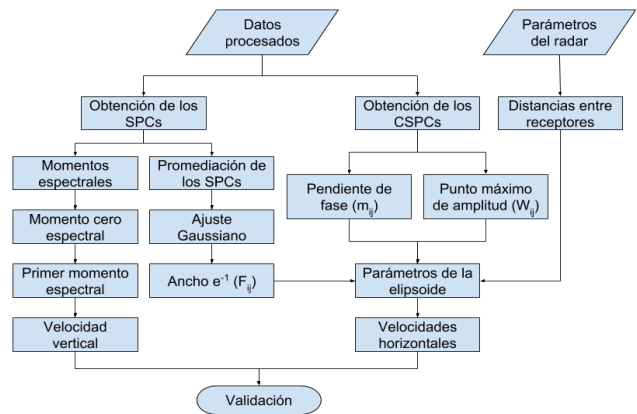


Figura 5. Diagrama de flujo detallado para la estimación de vientos

La distancia entre los receptores es un tema relevante, puesto que estos datos ingresan en el procesamiento para la estimación de los parámetros de la ecuación de la elipsoide y la estimación de las velocidades horizontales. La distancia entre los receptores de CLAIRE contiene una distancia entre receptores de 1.75 m en forma de triángulo equilátero. Un valor aproximado a la mitad del diámetro del transmisor. La Figura 6 muestra las distancias entre los tres receptores que forman un triángulo equilátero.

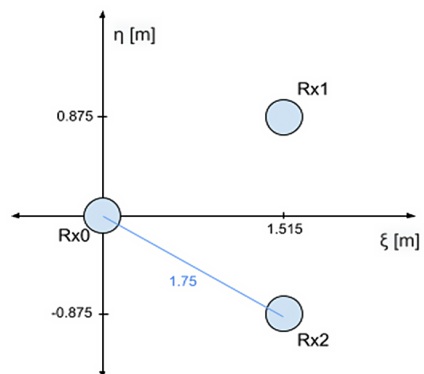


Figura 6. Distancias entre antenas receptoras de CLAIRE

Para la estimación de la velocidad vertical, se debe obtener el primer momento espectral de la señal. Los momentos espectrales presentan el algoritmo mostrado en la Figura 7. Con este método se puede obtener la potencia, velocidad radial y varianza de la señal aleatoria que se ingresa. Esta técnica también puede ser usada para obtener la velocidad de caída de hidrometeoros para la precipitación.

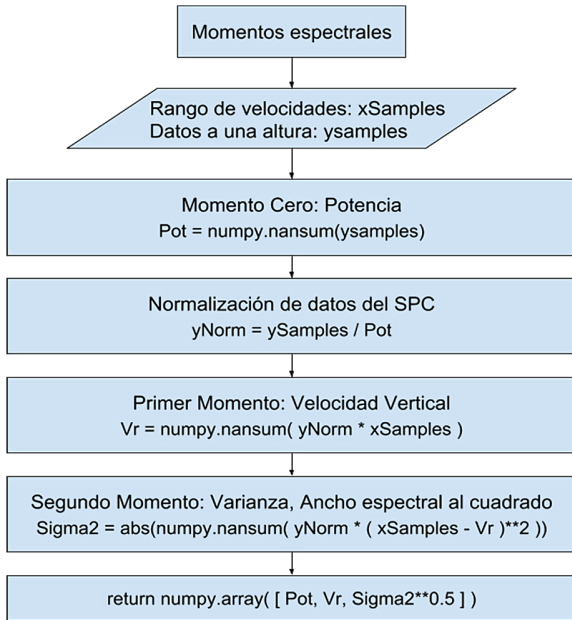


Figura 7. Diagrama de flujo para la estimación de momentos espectrales en Python

2.4.2. Algoritmo de factor de reflectividad

Para obtener el factor de reflectividad Z , la intensidad de lluvia y la velocidad de caída de lluvia. Los datos de entrada para el procesamiento de la estimación de precipitación deben ser los datos del SPC y la constante del radar.

Para obtener las estimaciones del factor de reflectividad e intensidad de lluvia es necesario hallar la distribución del tamaño de gotas N y el diámetro de hidrometeoros. Como se muestra en las siguientes ecuaciones:

$$Z(D) = \int_{D_{\min}}^{D_{\max}} D^6 N(D) dD$$

$$R(D) = 6 \times 10^{-4} \pi \int_{D_{\min}}^{D_{\max}} D^3 N(D) v_c(D) dD$$

La Figura 8 muestra el diagrama de flujo detallado para la estimación de precipitación. Del bloque de datos se puede obtener la relación velocidad diámetro. Éstas en relación al desplazamiento Doppler. La sección equivalente σ depende del diámetro del hidrometeoro D , la longitud de onda λ y el índice de refracción del agua $|Km|^2$.

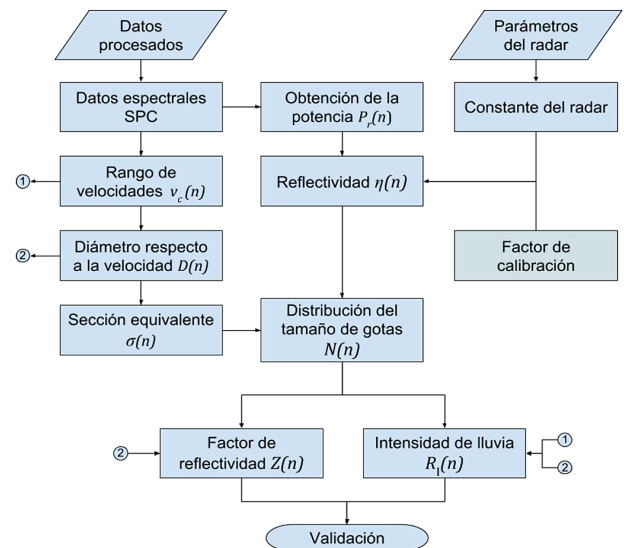


Figura 8. Diagrama de flujo detallado para la estimación de precipitación

3. RESULTADOS

Una vez implementados los algoritmos en Signal Chain y con el radar CLAIRE operando en el Observatorio de Huancayo, se comenzó a procesar las señales obtenidas. Los primeros resultados de vientos se pueden observar en la Figura 9 que contiene cuatro gráficas: velocidad zonal [m/s], velocidad meridional [m/s], velocidad vertical [m/s] y el SNR [dB]. La Figura 10 muestra los resultados de precipitación y contiene tres imágenes: Factor de reflectividad Z [dBZ], intensidad de lluvia R [mm/hr], velocidad de caída [m/s].

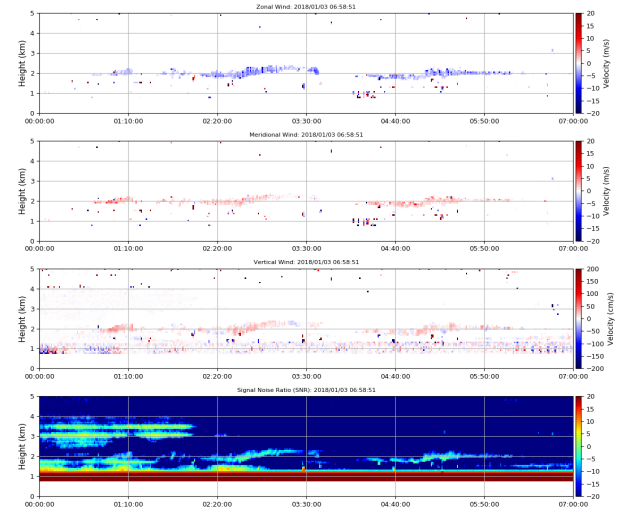


Figura 9. Resultados del procesamiento de vientos el día 03/01/2018

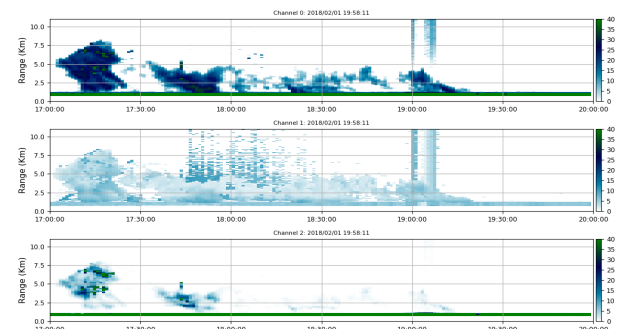


Figura 10. Resultados del procesamiento de vientos el día 01/02/2018

4. CONCLUSIONES Y RECOMENDACIONES

En este compendio se buscó resumir la tesis que fue desarrollada en el Observatorio de Jicamarca por el autor. Se llegó a conseguir resultados dentro del rango esperado. Sin embargo, aún es necesaria la implementación de estos algoritmos en el sistema de CLAIRE. Además actualizar Signal Chain con los cambios realizados en este proyecto. También, se observó problemas en los datos de CLAIRE como interferencia y Clutters que afectan su calidad del procesamiento. En el futuro se espera solucionar el problema de la interferencia y la calidad del procesamiento a tiempo real y que sea mostrado en línea para que la comunidad científica pueda realizar sus estudios pertinentes.

5. AGRADECIMIENTOS

Se agradece al Instituto Geofísico del Perú que dio inicio a este proyecto, al equipo del Radio Observatorio de Jicamarca que diseñó, desarrolló e implementó el proyecto de Radar UHF.

Así mismo al Doctor Danny Scipión Castillo y al grupo de ingenieros y técnicos del Radio Observatorio de Jicamarca que apoyaron en la implementación de este proyecto.

6. BIBLIOGRAFÍA

[1] B.H. Briggs. (1984). The analysis of spaced sensor records by correlation techniques. Adelaide, Australia: Handbook for MAP, volume 13, 166-185.

[2] B.H. Briggs y R.A. Vincent. (1992). Spaced-antenna analysis in the frequency domain. Adelaide, Australia: Radio Science, volume 27, 117-129.

[3] D. Atlas, R.C. Srivastava y R.S. Sekhon. (1973). Doppler Radar Characteristics of Precipitation at Vertical Incidenc. University of Chicago, Reviews of Geophysics and Space Physics, volume 11, 1-35.

[4] R.J. Doviak y R. Zrníc. (1993). Doppler radar and weather observations.

IMPLEMENTACIÓN DE SISTEMA DE RADAR IONOSONDA EN RED PITAYA

**Implementation of an ionosonde radar system using red pitaya*

Alejandro Jesús Castro Ortecho

Pontificia Universidad Católica del Perú
Facultad de Ciencias e Ingeniería – Sección
Electricidad y Electrónica
alejandro.castro@jro.igp.gob.pe

Asesor: Ing. Joaquín Verástegui

Radio Observatorio de Jicamarca - IGP

RESUMEN

El presente trabajo consistió en el diseño de un sistema de radar ionosonda basado en red pitaya tomando como base el trabajo realizado por Nina Fuchs. Se realizó un sistema que transmite una onda modulada con un sweep de frecuencias desde 1 MHz hasta 25 MHz, teniendo la posibilidad de modular la señal portadora con un código variable durante cada step del sweep. Paralelamente este sistema recibe la data, la procesa, la almacena a 100 KHz y la transmite a través de un puerto Gigabit Ethernet usando el protocolo de comunicación TCP. Para lograr lo descrito anteriormente se desarrolló el sistema completo usando el software Vivado de Xilinx utilizando sus IPs así como desarrollando los controladores necesarios para su correcto funcionamiento usando el lenguaje de descripción de hardware Verilog. Se realizó el desarrollo de la interfaz con el usuario usando programación en C. Y por último se realizó las pruebas respectivas de cada parte para asegurar su correcto funcionamiento.

Palabras clave: ionosonda, verilog, C, fpga, procesador, transmisión, recepción, TCP.

ABSTRACT

This work presents the design of an ionosonde radar system using red pitaya taking as a reference the work made by Nina Fuchs. The system transmits a modulated signal with a frequency sweep from 1 MHz to 25 MHz, and letting the user modulates the signal with a variable code during each step of the sweep. In parallel the system receives the data, processes them, stores it at 100 kHz and transmits it through a Gigabit Ethernet port using the TCP communication protocol. To accomplish all the functionality described, I developed the entire system by using the IPs provided by the Vivado software as well as designing the necessary controllers to ensure the correct performance of the entire system using the hardware description language Verilog. Later I developed a software user interface using the C programming language. And finally I tested each part of the system to ensure their correct functionality.

Keywords: [ionosonde, verilog, C, fpga, processor, transmission, reception, TCP].

2.1.2. Sistema de interfaz entre el fpga y el procesador

El sistema de interfaz entre el fpga y el procesador Memory Access está conformado por distintos bloques de hardware que se encargan de leer los datos enviados por el procesador como el IPP, ancho de pulso, código, ancho de código, comandos y enviar los distintos flags necesarios al procesador tal como se muestra en la Figura 3.

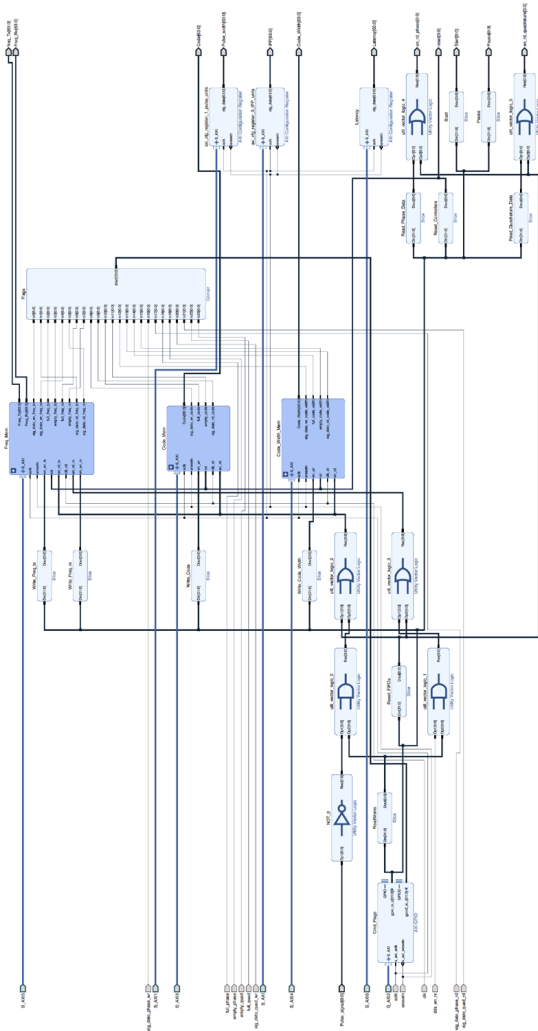


Figura 3: Diagrama de bloques interno del bloque Memory Access

2.1.3. Sistema de recepción, procesamiento y almacenamiento

El sistema de recepción, procesamiento y almacenamiento está compuesto por un DDS de frecuencia variable sincronizado con la señal modulada en amplitud, dos bloques de procesamiento tanto para la fase como para la cuadratura, dos módulos sampler que se encargan de muestrear la señal procesada a una frecuencia de 100 KHz y dos memorias, que se comunican con el programa interfaz mediante un AXI_GPIO dual tal como se muestra en la Figura 4.

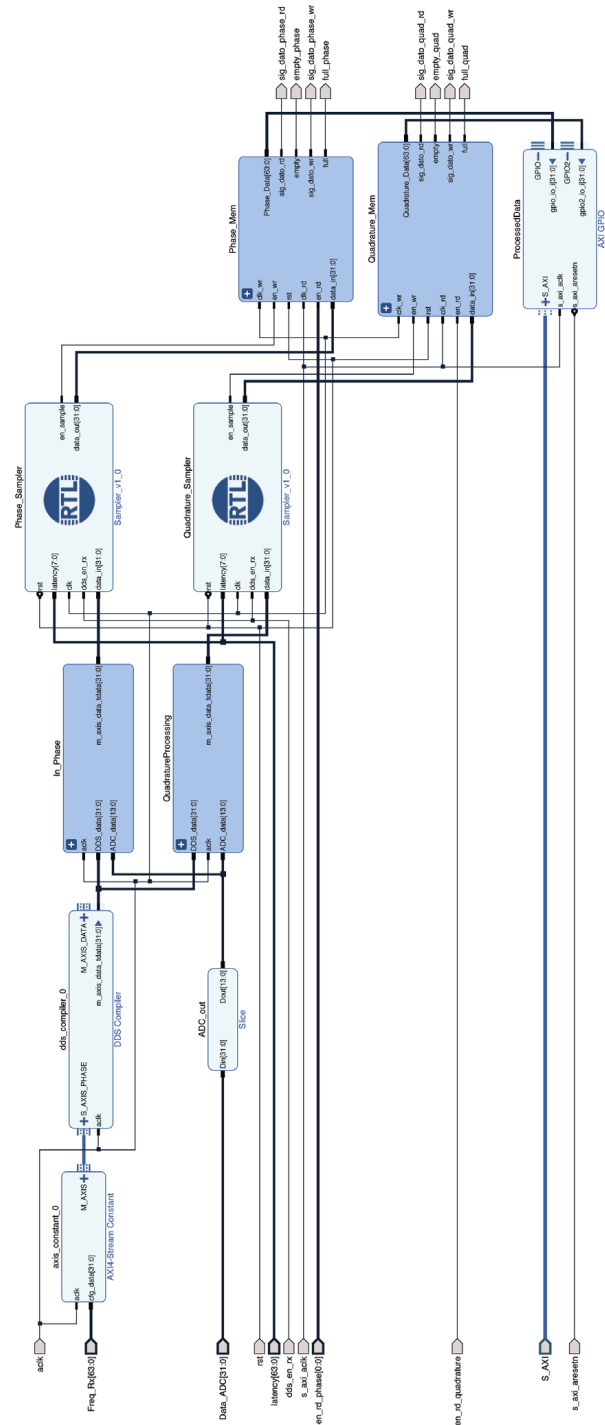


Figura 4: Diagrama de bloques interno del bloque Receiver

Cada bloque de procesamiento está conformado por un bloque slice que obtiene las señales fase y cuadratura a partir de la salida del DDS, dos CIC_Compiler decimadores, uno con 3 etapas y una constante de submuestro de 25 y el otro con 5 etapas y una constante de submuestro de 5. Adicionalmente cuenta con un filtro FIR_Compiler con frecuencia de corte de 100 KHz para eliminar completamente la portadora. La Figura 5 muestra la interacción entre dichos bloques para la señal de cuadratura.

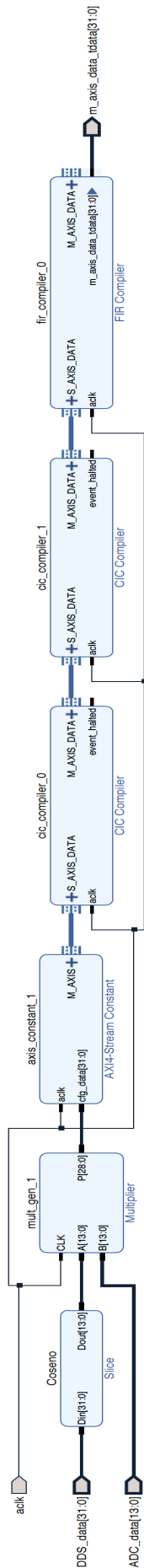


Figura 5: Diagrama de bloques interno del bloque de procesamiento Quadrature Processing.

2.2. Diseño del software del sistema

El software de interface que corre en el procesador de red pitaya fue programado en C. Fue desarrollado para sincronizarse con el hardware desarrollado en el fpga, es decir su ejecución no depende del *timing* de las señales. La ejecución del mismo sigue el diagrama de flujo mostrado en la Figura 6.



Figura 6: Diagrama de flujo resumido del programa de interface del sistema radar

3. RESULTADOS

- Se logró un sweep de frecuencia y una modulación en fase con código totalmente configurable por el usuario mediante el uso de archivos txt.

- Se logra sincronizar el sistema de recepción con el de transmisión mediante el uso de una señal especial llamada `dds_en_rx` que es generada por el modulador en amplitud.
- Se logró poder configurar el sistema para que funcione teniendo en cuenta la demora de los bloques de procesamiento (latency).
- Se logró realizar transmisiones de paquetes TCP desde red pitaya hasta un servidor sin pérdida de datos.

4. DISCUSIÓN

Para diseñar hardware en fpgas se recomienda usar la última versión de los programas del fabricante siendo en este caso el software Vivado para así tener acceso a todos los ips más recientes que pueden ser útiles en el proyecto. Por otro lado para desarrollar software de interfaz y adquisición de data se recomienda usar C, dado que el procesador de red pitaya funciona a 600 MHz y el fpga a 125MHz, lo que hace que sea necesario utilizar lenguajes rápidos. Esto se corroboró ya que en un inicio se desarrolló la interfaz en Python, pero la lentitud de este lenguaje hacía imposible la adquisición de data.

5. CONCLUSIONES

Se mejoró el sistema de transmisión.

Se implementó el sistema de memorias que permiten cambiar tanto la frecuencia, el código como su ancho durante cada IPP.

Se implementó el sistema receptor con bloques de procesamiento y un muestreador a 100 KHz que almacena y posteriormente envía la data al programa interfaz.

Se programó una interfaz en C que lee la data procesada y se envía a un servidor usando TCP.

Se revisó de cada parte, en especial la sincronía tanto en los bloques del fpga como entre el programa interfaz y el fpga para hacer las modificaciones necesarias que garanticen la robustez del sistema.

Se realizó simulaciones de cada parte por separado para asegurar su correcto funcionamiento.

6. RECOMENDACIONES

Se recomienda cambiar las memorias basadas en FIFOs a memorias basadas en RAMs, esto permite disminuir el tiempo de inicio de cada ejecución del programa interfaz ya que no se tendría que escribir de nuevo dichas memorias.

Se recomienda hacer un testbench lógico global modelando el programa interfaz en Verilog lo cual no se hizo debido a la complejidad del mismo y a que no se disponía del tiempo requerido.

Se recomienda contar con una computadora potente cuando se quiera realizar desarrollo de hardware extenso dado que el proceso de síntesis e implementación que se debe realizar cada vez que se desea visualizar un cambio hecho en el diseño del hardware toma demasiado tiempo, incrementándose dicho tiempo conforme la descripción de hardware crece, lo cual ralentiza considerablemente el proceso de desarrollo y testeo.

Se recomienda testear el sistema transmitiendo mediante una antena y recibiendo por la misma u otra antena o en su defecto, simular el efecto de la ionósfera mediante un circuito atenuador y con ruido para visualizar el funcionamiento real del sistema desarrollado.

Se recomienda utilizar una señal GPS para hacer funcionar los DDS ya que esta señal permite tener señales moduladas mucho más sincronizadas.

7. AGRADECIMIENTOS

Deseo agradecer al Instituto Geofísico del Perú a través del ROJ por haberme permitido ser partícipe de su programa de prácticas pre profesionales. También me gustaría expresar mi más sincero agradecimiento al Ing. Joaquín Verástegui, mi asesor de proyecto, por el apoyo que me brindó durante estos meses esclareciéndome muchas dudas acerca de como debería ser el sistema y como funcionan los radares con los que cuenta el ROJ para tener un base sobre la que estructurar el desarrollo de mi proyecto, así como por muchos otros consejos.

8. BIBLIOGRAFÍA

Xilinx, Vivado Design Suite (2016). CIC Compiler v4.0. LogiCORE IP Product Guide, pp 9-19.

Xilinx, Vivado Design Suite (2015). FIR Compiler v7.2. LogiCORE IP Product Guide, pp 8-69.

Xilinx, Vivado Design Suite (2017). DDS Compiler v6.0. LogiCORE IP Product Guide, pp 8-29.

Xilinx, Vivado Design Suite (2017). FIFO Generator v13.1. LogiCORE IP Product Guide, pp 66-148.

IMPLEMENTACIÓN Y ANÁLISIS DE ALGORITMOS DE FORMACIÓN DE IMÁGENES PARA RADARES DE APERTURA SINTÉTICA DE ESTACIÓN TERRENA (GB-SAR) DE MONITOREO DE DESLIZAMIENTOS

**Implementation and analysis of Ground Based Synthetic Aperture Radar (GB-SAR) Imaging Algorithms for Landslide Monitoring*

Luis De La Cruz Painado

Universidad Nacional de Ingeniería
Facultad de Ingeniería Eléctrica y Electrónica
luis.delacruz@jro.igp.gob.pe

Asesor: Dr. Marco A. Milla Bravo

Investigador Principal – Radio Observatorio de Jicamarca - IGP

RESUMEN

El propósito del presente trabajo de tesis es realizar un estudio comparativo de 3 algoritmos de formación de imágenes para radares de Apertura Sintética de Estación Terrena (GB-SAR): Frequency Domain Back Projection (FDBP), Range Migration Algorithm (RMA) y Discrete Linear Inverse Problem (DLIP). Este estudio consiste en implementar los 3 algoritmos de formación de imágenes anteriormente mencionados, usando tanto datos simulados como reales; para posteriormente compararlos usando los criterios de calidad de reconstrucción de la imagen y coste computacional. Los resultados de comparación servirán para decidir el algoritmo a emplear en las primeras etapas del desarrollo de un sistema de monitoreo de deslizamientos con radares SAR, realizado en el Instituto Geofísico del Perú en su Radio Observatorio de Jicamarca.

Palabras clave: GB-SAR, Algoritmos de formación de imágenes, Algoritmo FDBP, Algoritmo RMA, Algoritmo DLIP, Monitoreo de deslizamientos.

ABSTRACT

This thesis work describes the development and implementation of 3 imaging algorithms using Ground Based Synthetic Aperture Radar (GB-SAR): Frequency Domain Back Projection (FDBP), Range Migration Algorithm (RMA), and Discrete Linear Inverse Problem (DLIP). Afterward, all the algorithms will be compared using simulated and real data. The comparison criteria will be the quality of image reconstruction and the computational cost. Finally, an algorithm will be chosen to be used in the first stages of a Landslide Monitoring System using SAR radars, which has been developed at the Geophysical Instituto of Peru at its scientific facility Jicamarca Radio Observatory.

Keywords: GB-SAR, Imaging Algorithms, FDBP Algorithm, RMA Algorithm, DLIP Algorithm, Landslide

1. INTRODUCCIÓN

El Instituto Geofísico del Perú (IGP) viene desarrollando, en el Radio Observatorio de Jicamarca (ROJ), el proyecto “Sistema de Monitoreo de Deslizamientos usando Interferometría de Radar” [1], el cual consiste en realizar el monitoreo de deslizamientos empleando un Radar de Apertura Sintética de Estación Terrena (GB-SAR). El presente trabajo se enfocará en la parte inicial del proyecto el cual es la “Formación de las imágenes de reflectividades”, para ello se realizará un estudio comparativo de 3 Algoritmos de Formación de Imágenes para radares SAR que se podrían usar para este propósito.

2. DESARROLLO

2.1. Descripción del sistema de adquisición de datos y simulaciones

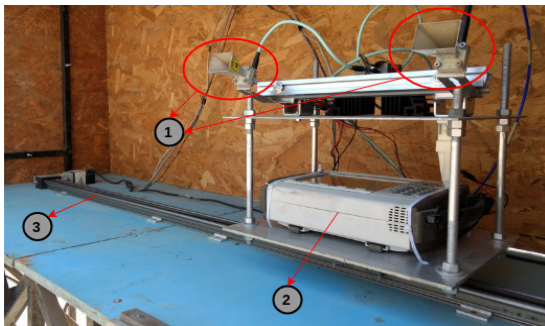


Figura 1. Radar GB-SAR construido en la institución IGP-ROJ, se muestran sus partes principales: 1) Antenas de Bocina, 2) Analizador Vectorial de Redes (VNA) y 3) Sistema de posicionamiento lineal.

2.1.1. Principio de Operación

Un radar GB-SAR con apertura lineal presenta las siguientes partes principales: un par de antenas (una de transmisión y una de recepción), un sistema de posicionamiento lineal, un Analizador Vectorial de Redes (VNA), un amplificador de potencia (PA) y una computadora. En la Figura 1 se muestra el radar GB-SAR construido en la institución IGP-ROJ el cual se usa para la adquisición de los datos.

La función del radar GB-SAR es obtener data cruda o Raw Data del blanco a reconstruir, el blanco en este caso es el cerro propenso a deslizamiento. La data cruda es una función de 2 variables: posición y frecuencia, el cual se obtiene captando el eco de la señal transmitida por cada frecuencia y posición en el riel usado. El tipo de señal transmitida es Stepped Frequency Continuous Wave (SF-CW). El Raw Data finalmente conforma el dato de entrada a cada uno de los Algoritmos de Formación de Imágenes.

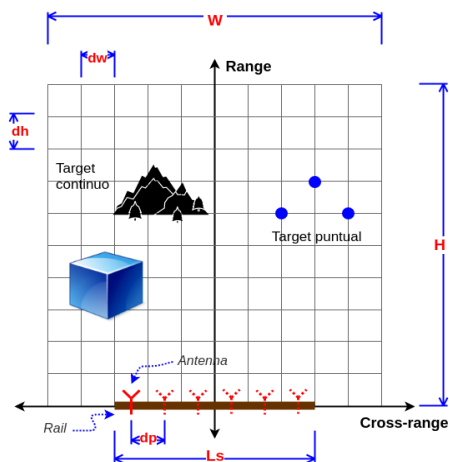


Figura 2: Geometría usada en la implementación de los algoritmos.

Otro aspecto importante a definir al inicio es la geometría del sistema de formación de imágenes GB-SAR, el cual hace referencia a 3 aspectos: el plano de coordenadas a usar, la ubicación del radar GB-SAR dentro de este plano y la ubicación de la escena de la imagen a reconstruir dentro de este plano. En la Figura 2 se muestra la Geometría que se usará.

2.1.2. Simulación de datos

Para facilitar la prueba de los algoritmos y realizar la comparación entre ellos, se obtuvo de manera teórica la data cruda o Histórico de Fase, para ello se usó la siguiente expresión [2]:

$$S(u_k, f_i) = \sum_{j=1}^N a_{tj} e^{-j4\pi \frac{f_i}{c} \sqrt{x_{tj}^2 + (u_k - y_{tj})^2}}$$

Donde (x_{tj}, y_{tj}) representa la posición y a_{tj} la reflectividad de cada blanco puntual. Sin perder generalidad se asume $a_{tj} = 1$ para todos los casos de obtención de Histórico de Fase teórico dentro de este trabajo.

2.2 Implementación de Algoritmos de Formación de Imágenes SAR

2.2.1. Primer algoritmo: Frequency Domain Back Projection (FDBP)

Este algoritmo se basa en las siguientes ecuaciones: [3]

$$I_n = \frac{1}{N_f N_p} \sum_k R_{n,k}^2 F_n^{(k)}$$

$$F_n^{(k)} = \sum_i S_{k,i} e^{j \frac{4\pi f_i}{c} (R_{n,k} - R_o)}$$

Donde $R_{n,k}$ es la distancia entre el punto “n” y la posición “k” del riel (u_k) , R_o es una constante que toma en cuenta el retraso (delay) de los cables, f_i es la frecuencia “i-ésima” y $S_{k,i}$ es el histórico de fase. La magnitud de la imagen compleja representa la reflectividad.

Una forma práctica de poder implementar el algoritmo y hallar I_n es siguiendo los pasos mostrados en la Figura 3. A continuación se explican estos pasos, en primer lugar se definen los parámetros mecánicos, eléctricos y electrónicos a usar, con ellos se obtiene el Raw Data, luego se realiza una compresión en rango mediante una 1D-IFFT en el eje de la frecuencia a los datos, a continuación se realiza una interpolación lineal a toda la imagen por cada posición del riel, finalmente se suman todas las imágenes interpoladas entre sí para obtener la imagen reconstruida I_n .

2.2.2. Segundo algoritmo: Range Migración (RMA)

Este algoritmo se caracteriza por trabajar en el dominio de la frecuencia y por requerir una interpolación 1D en particular llamada “Interpolación STOLT”.



Figura 3: Pasos del Algoritmo FDBP

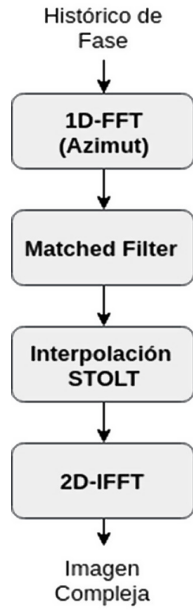


Figura 4: Pasos del Algoritmo RMA

Los pasos se muestran en la Figura 4, los cuales se basaron de [4] y se explican a continuación. Una vez obtenido el Raw Data, se llevan los datos al dominio de la frecuencia mediante la operación 1D-FFT en el eje de azimut. Después de ello se realiza un filtrado tipo Matched Filter el cual consiste en variar la fase de la señal, a continuación, se efectúa una interpolación del tipo Stolt el cual consiste en realizar un cambio de coordenadas y una interpolación lineal. Finalmente se efectúa la operación 2D-IFFT, el cual lleva la señal del dominio de la frecuencia al dominio del espacio.

2.2.3. Tercer Algoritmo: Discrete Linear Inverse Problem (DLIP)

Este algoritmo se basa en la idea de aproximar el sistema de un Radar de Apertura Sintética para formación de imágenes (Imaging SAR) a un sistema lineal, y así poder reconstruir las imágenes aplicando Teoría de Álgebra Lineal y de Problemas de Inversión.[5]

Específicamente el algoritmo se basa en resolver la siguiente ecuación lineal y hallar χ :

$$[E_{sca}] = [A][\chi]$$

Donde:

- $[E_{sca}]$ es el Histórico de Fase mostrado en un vector columna de tamaño LS , en donde L es el número de posiciones del riel y S es el número de frecuencias usadas.
- $[\chi]$ es el vector de reflectividades, mostrados en un vector columna de tamaño N , en donde N es el número de píxeles de la escena de la imagen.
- $[A]$ es la matriz que identifica al sistema. Tiene dimensión igual a $(LS \times N)$.

La matriz característica del sistema $[A]$ presenta la siguiente expresión aproximada:

$$A(\bar{r}_l, k_s, \bar{r}_n) = e^{[-j4\pi \frac{k_s}{c} |\bar{r}_n - \bar{r}_l|]}$$

Donde \bar{r}_l representa la posición del riel, k_s las frecuencias \bar{r}_l y la posición de cada pixel de la imagen.

La solución estará dada por la siguiente expresión:

$$[\chi] = [A]^\dagger [E_{sca}]$$

Donde $[A]^\dagger$ es la pseudoinversa de $[A]$, el cual posee una de las siguientes expresiones dependiendo de sus dimensiones matriciales:

1. Sistemas subdeterminados (LS < N):

$$[A]^\dagger = A^T (AA^T)^{-1}$$

2. Sistemas sobredeterminados (LS > N):

$$[A]^\dagger = (AA^T)^{-1} A^T$$

3. Sistemas cuadráticos (LS = N):

$$[A]^\dagger = A^{-1}$$

3. RESULTADOS

3.1. Pruebas y comparación con datos simulados

Para las pruebas con datos simulados, se define un blanco en forma de T, tal como se ve en la Figura 5. Los parámetros mecánicos, eléctricos y geométricos usados se muestran en la Tabla 1. Su respectiva reconstrucción empleando los Algoritmos FDBP, RMA y DLIP se muestra en la Figura 6.

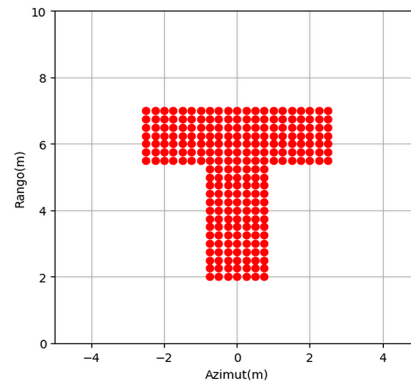


Figura 5: Blanco en forma de T usado para comparar los 3 Algoritmos de Formación de Imágenes implementados con datos simulados.

PARÁMETRO	VALOR
Frecuencia central: fc	15 GHz
Ancho de banda: BW	600MHz
Número de frecuencias: Nf	41
Apertura sintética: Ls	1.2 m
Ángulo de visión: theta	75°
Número de posiciones: Np	238
Número de píxeles en el eje x: Nx	41
Número de píxeles en el eje y: Ny	41
Longitud en el eje x: Lx	10 m
Longitud en el eje y: Ly	10 m
Resolución de la imagen: dx, dy	0.25 m

Tabla 1: Parámetros eléctricos, mecánicos y geométricos usados en la implementación y comparación de 3 Algoritmos de Formación de Imágenes SAR

Después de la reconstrucción se procedió a la comparación de los 3 algoritmos. Los parámetros de comparación que se consideraron fueron: calidad de reconstrucción de la imagen y coste computacional. La calidad de reconstrucción de la imagen se obtiene calculando los errores de la imagen reconstruida respecto a la imagen del blanco original. El coste computacional por su parte se obtiene del tiempo de procesamiento del algoritmo en un computador, las características del ordenador

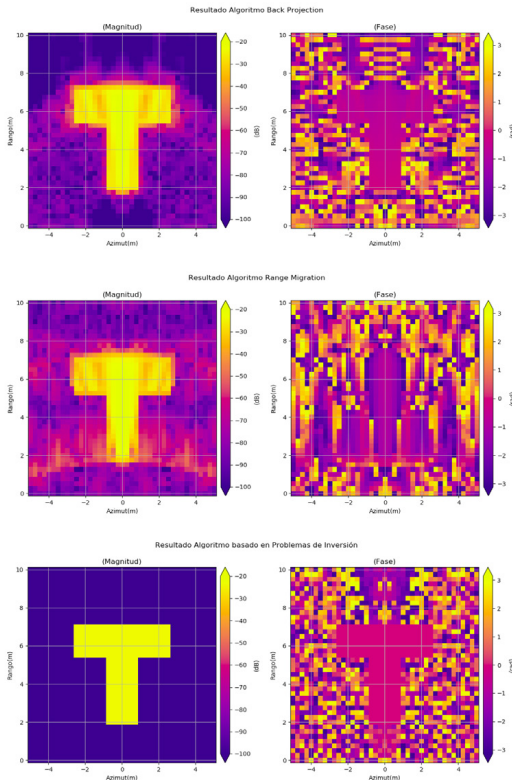


Figura 6: Imagen reconstruida en magnitud y fase de un blanco en forma de T usando los algoritmos: FBDP (Parte superior), RMA (Parte Media) y DLIP (Parte Inferior).

usado son: procesador Core i5 y 8 GB de memoria RAM. En la Tabla 2 se muestran los valores numéricos de estos dos parámetros de comparación. En ella se puede observar que el Algoritmo FBDP presenta el menor coste computacional, mientras que el algoritmo DLIP presenta los valores más bajos en los distintos tipos de errores. Evaluando valores adecuados para ambos parámetros: bajo valor en el coste computacional y en los errores de magnitud y fase; se llega a determinar que el Algoritmo FBDP es el que cumple estos requerimientos.

PARÁMETRO	FBDP	RMA	DLIP
Error cuadrático medio(L2 norm)	9.26E-05	1.61E-04	2.55E-28
Raíz cuadrada del error cuadrático medio	0.00962	0.0127	1.60E-14
Valor máximo del error(max norm)	0.0484	0.0605	1.55E-13
Error absoluto medio(L1 norm)	0.0035	0.00494	5.68E-15
Error cuadrático medio de la fase(rad)	0.757	1.335	2.05E-14
Tiempo de procesamiento(s)	0.106	0.149	113.36

Tabla 2: Cuadro comparativo para un blanco en forma de T. Se muestran parámetros de la calidad de reconstrucción de la imagen y coste computacional.

3.2. Pruebas con datos reales

Las pruebas con datos reales se realizan para comprobar la correcta implementación de cada algoritmo. Debido al alto

coste computacional del Algoritmo DLIP, este no se usa para estas pruebas.

El blanco en este caso es una zona del cerro Pucruchacra ubicado en el km 95 de la Carretera Central, en el distrito de San Mateo, provincia de Huarochirí, Lima (ver Figura 7); el cual presentó indicios de deslizamiento en años pasados. Los parámetros eléctricos, mecánicos y geométricos que se usan en esta prueba se muestran en la Tabla 3.



Figura 7: Cerro Pucruchacra(vista aérea) que será usado como blanco para las pruebas con datos reales.

PARÁMETRO	VALOR
Frecuencia central: f_c	15.55 GHz
Ancho de banda: BW	100 MHz
Número de frecuencias: N_f	1001
Apertura sintética: L_s	1.4 m
Ángulo de visión: θ	75°
Número de posiciones: N_p	178
Longitud en el eje x: L_x	400 m
Longitud en el eje y: L_y	650 m
Posición inicial eje y: y_i	300 m
Resolución de la imagen: dx, dy	1 m

Tabla 3: Parámetros eléctricos, mecánicos y geométricos usados en las pruebas con datos reales.

En la Figura 8 se muestra las imágenes reconstruidas (en magnitud) para los Algoritmos FBDP y RMA.

Se puede observar en ambas figuras la presencia de surcos concéntricos desde 300 hasta 500m en el eje de rango, con mayor intensidad en el Algoritmo FBDP, estos surcos también se presentan en el blanco original (Ver Figura 7 encerrados en una elipse roja), con lo cual se corrobora la correcta reconstrucción empleando ambos algoritmos. Respecto al coste computacional, el Algoritmo FBDP posee un tiempo de procesamiento de 7s aproximadamente, mientras que el Algoritmo RMA demora 168s; con esto se demuestra el menor coste computacional del Algoritmo FBDP empleando datos reales. Una diferencia que se puede apreciar comparando ambas imágenes reconstruidas es que la imagen producto del Algoritmo FBDP presenta mayores regiones con alto SNR que el Algoritmo RMA. Esta característica de tener mayores regiones con alto SNR permite posteriormente realizar un análisis de deslizamientos en más regiones de la imagen, lo cual es una ventaja para el algoritmo FBDP.

4. CONCLUSIONES

Se logró implementar satisfactoriamente 3 algoritmos de formación de imágenes para radares GB-SAR: FBDP, RMA y DLIP,

los cuales se probaron con datos simulados los 3 algoritmos y con datos reales 2 de ellos: FDBP y RMA.

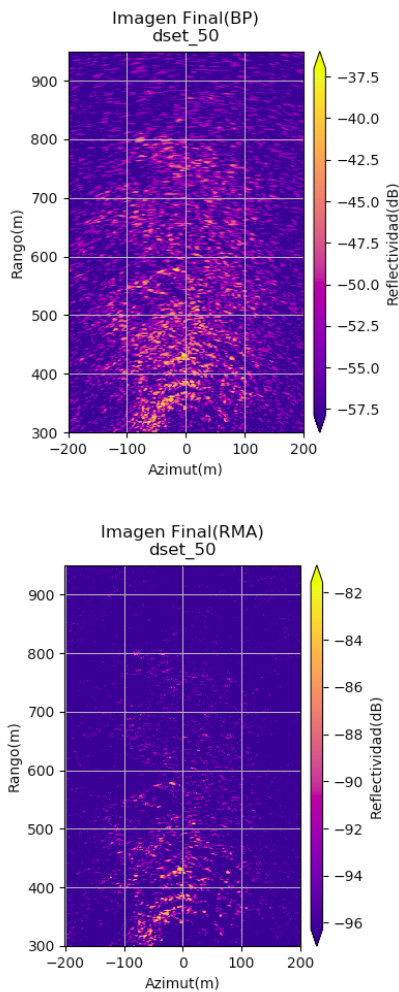


Figura 8: Imagen reconstruida en magnitud de un blanco real, usando los algoritmos: FDBP (Parte superior) y RMA (Parte Inferior).

De la comparación en general entre los algoritmos FDBP, RMA y DLIP con datos simulados y reales, el algoritmo FDBP es el que presenta mejores resultados debido a las siguientes razones: menor coste computacional, calidad media de reconstrucción de la imagen y alto SNR en más regiones de la imagen reconstruida.

5. AGRADECIMIENTOS

Al Instituto Geofísico del Perú, a través del Radio Observatorio de Jicamarca, por permitirme formar parte de y realizar mi tesis de grado en el proyecto: "Sistema de monitoreo de deslizamientos usando interferometría de radar"

Al Dr. Marco Milla y al Ing. André Florentino por su apoyo constante y motivación.

A todo el personal del área "CIELO" por los consejos y orientaciones.

Finalmente, al Fondo Nacional de Desarrollo Científico y Tecnológico (FONDECYT) por su apoyo bajo el convenio de adjudicación No. 099-2014-FONDECYT-DE.

6. BIBLIOGRAFÍA

[1] A. Florentino, S. Charapaqui, C. De La Jara, and M. Milla. Implementation of a ground based synthetic aperture radar (GB-SAR) for landslide monitoring: system description and preliminary results. In 2016 IEEE XXIII International Congress on Electronics,

Electrical Engineering and Computing (INTERCON), pages 1–6, Aug 2016.

[2] Gregory L. Charvat, "Small and Short-Range Radar Systems", pp. 140-141, 2014.

[3] M. Pieraccini, G. Luzi, and C. Atzeni. Terrain mapping by ground-based interferometric radar. IEEE Transactions on Geoscience and Remote Sensing, 39(10):2176–2181, Oct 2001.

[4] R.S. Goodman W.G. Carrara and R.M. Majewski. Spotlight Synthetic Aperture Radar Signal Processing Algorithms. Artech House, 1995.

[5] L. Li, W. Zhang, and F. Li. Derivation and discussion of the SAR migration algorithm within inverse scattering problem: Theoretical analysis. IEEE Transactions on Geoscience and Remote Sensing, 48(1):415–422, Jan 2010.

INTEGRACIÓN DEL RADAR BLTR CON LAS APLICACIONES REALTIME Y LA BASE DE DATOS MADRIGAL

**Integration of the BLTR radar with the RealTime application and the Madrigal database*

Geordano Wilfredo Chagua Hidalgo

Universidad Nacional de Ingeniería
Facultad de Ciencias
geordano.chagua@jro.igp.gob.pe

Asesor: Juan Carlos Espinoza

Radio Observatorio de Jicamarca - IGP

RESUMEN

En el presente trabajo describe el procesamiento de datos adquiridos por el radar perfilador de vientos BLTR (boundary layer and tropospheric radar). Para tal propósito implementó un script en lenguaje de programación Python para procesar los datos del radar que almacena los valores de la velocidad de los vientos en las siguientes direcciones: zonal, vertical y meridional.

Iniciaremos con la comprensión de la estructura de la librería Signal Chain para posteriormente escribir el script de procesamiento, luego se pasará a testear el script y obtener los gráficos a partir de los archivos almacenados por la computadora del radar para finalmente poder subir los datos obtenidos a la base de datos Madrigal.

Palabras clave: radar, script, Signal Chain, python

ABSTRACT

In the present work, the data processing of the data acquired by the BLTR (boundary layer and tropospheric radar) is presented. For this purpose, we have to implement a script in Python to process the data acquired by the radar and stored wind values in the following directions: zonal, vertical, and meridional.

We will start by understanding the structure of the Signal Chain library to write the processing script, and then test it to obtain the graphics and finally, upload them into the Madrigal database.

Keywords: radar, script, signal chain, python

1. INTRODUCCIÓN

En el país existen lugares en donde los microclimas favorecen la presencia de vientos permanentes, y pueden ser aprovechados para aplicaciones de la energía eólica especialmente en territorios de la costa, por tal motivo el estudio de los vientos cobra importancia en nuestro país; sin embargo, existe la probabilidad de encontrar en otros sitios estos recursos.

El propósito de este proyecto es implementar un script usando los módulos de la biblioteca signal chain que realice la lectura, procesamiento y posterior gráfico de los datos registrados con el perfilador de vientos BLTR (Boundary Layer and Tropospheric Radar) ubicado en Huancayo.

Perfilador de vientos

Un perfilador de vientos es un instrumento de monitoreo de la atmósfera de manera remota y que usa ondas de radio para detectar velocidad y dirección del viento a varias elevaciones sobre el terreno, información valiosa en meteorología y a la aviación.



2. DESARROLLO

2.1. Esquema del proceso de trabajo

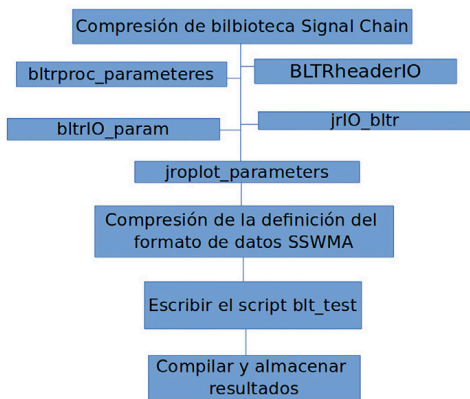


Figura 2 : Proceso de trabajo

2.2. Signal Chain

Signal Chain (SCh) es una biblioteca de código abierto para el procesamiento de señales adquiridas con radares científicos, desarrollada en el Radio Observatorio de Jicamarca, sede científica del Instituto Geofísico del Perú. SCh proporciona módulos para leer, escribir, procesar y trazar datos.

En SCh, cada algoritmo de procesamiento de señales se divide en bloques separados. Este enfoque se puede explicar en la siguiente imagen, donde el proceso de adquisición de una imagen digital se ilustra como una secuencia. El proceso se divide en cinco bloques diferentes. Cada uno de ellos juega un papel diferente. El primer paso es transformar la luz en una señal de voltaje con una matriz de sensores. A continuación, la señal analógica obtenida se amplifica con un bloque amplificador de bajo ruido y, a continuación, se convierte a digital a través de un ADC. El siguiente bloque es un DSP, que involucra muchos algoritmos de procesamiento de señales como el filtrado o el promediado. Finalmente, la última etapa es almacenar la imagen obtenida en una base de datos.

De manera análoga, para desarrollar bibliotecas SCh, cada algoritmo se divide en una secuencia de operaciones independientes, como la lectura de datos, el filtrado de interferencias o el trazado de resultados. Estos agentes tienen un alcance deliberadamente limitado y son tan genéricos como lo permite el algoritmo. El propósito es hacer que los módulos sean reutilizables en diferentes procesos y, de esta manera, evitar la repetición del código. Esta característica se ilustra en la imagen de abajo.

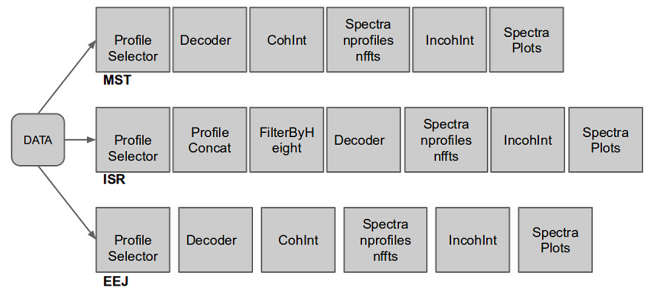


Figura 2 : Proceso de trabajo

2.2.1 Capas de Configuración

Cada experimento SCh se configura en cuatro pasos o capas básicas, que se describen a continuación.

1. Interfaz de usuario: La configuración se puede realizar con la interfaz gráfica (GUI) o un script de Python, que es la opción recomendada para un desarrollador.

2. Proyecto: A continuación, se debe crear un objeto Proyecto, que tendrá una identificación, un nombre y una breve descripción del experimento que se está procesando.

3. Unidades: Dentro del Proyecto, el usuario creará Unidades, que contendrán las diferentes operaciones o tareas que SCh realizará a los datos. Hay tres tipos principales de Unidades:

Unidad de lectura, estas unidades leen los datos adquiridos por el sistema de radar. Estos datos pueden ser voltajes o información espectral.

Unidad de escritura, una vez que se han procesado los datos, la información obtenida o los datos modificados se pueden almacenar en archivos que se pueden leer más adelante.

Unidad de procesamiento, contienen las diferentes operaciones matemáticas o algoritmos que se realizarán a los datos de un tipo específico. El tipo de operaciones que se pueden seleccionar depende del tipo de unidad de procesamiento que se está utilizando. Hay cuatro tipos principales de unidades de procesamiento: voltaje, espectros, correlaciones y parámetros.

4. Operaciones.- Finalmente, el usuario debe agregar a las Unidades todos los algoritmos u operaciones matemáticas. Hay dos tipos de operaciones:

Operación de procesamiento, algoritmos o cálculos para obtener información específica de los datos, como el cálculo de momentos, estimación de vientos o decodificación.

Operación gráfica, gráficos para visualizar los resultados de los algoritmos de procesamiento de señales.

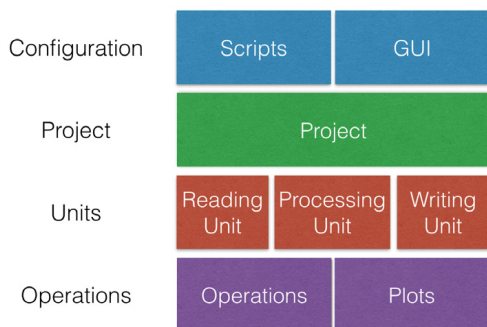


Figura 4: Operaciones de SChain

2.3. Formato SSWMA

El radar BLTR toma datos y los almacena en un formato binario de extensión SSWMA (Spaced Sensor Winds Measurements Analysis).

Cada archivo de datos está precedido por un encabezado de archivo (File Header) que contiene información general sobre el propio archivo de datos. Luego del encabezado del archivo hay una serie de registros, uno para cada período de adquisición de datos, que contienen los parámetros derivados del programa de análisis SSWMA. Los archivos de datos SSWMA producidos “en línea” se rotan las 24 horas.

2.3.1. Data

Los resultados siguen inmediatamente después del encabezado de la grabadora. Cada resultado se compone de un conjunto de parámetros calculados que se repiten para cada puerta de rango de datos. Los resultados para cada puerta de rango se organizan como se define a continuación. Nota

los campos de datos para los cuales los resultados no han sido o no pudieron calcularse se completan con un indicador de “valor incorrecto” El valor numérico del “valor malo” es -9999.

Offset	Size	Name	Format
0	4	Range (m)	int
4	4	Analysis status code	int
8	4	Zonal wind velocity (m/s) positive Eastwards	float
12	4	Meridional wind velocity (m/s) positive Northwards	float
16	4	Vertical wind velocity (m/s) positive upwards	float
20	4	Uncorrected zonal wind velocity (m/s)	float
24	4	Uncorrected meridional wind velocity (m/s)	float
28	4	Corrected fading time (s)	float
32	4	Uncorrected fading time (s)	float
36	4	Normalised time discrepancy (percentage)	float
40	4	Ellipse major axis length (m)	float
44	4	Ellipse axial ratio (major/minor)	float
48	4	Ellipse orientation (bearing in degrees measured clockwise from true north)	float
52	4	Sea scatter power calculated for this range (0 = no, 1 = yes)	int
56	4	Sea scatter removal algorithm applied to this range (0 = no, 1 = yes)	int
60	4*N	Receiver saturation (number of data points), one per acquired receiver channel	int
60+4N	8*N	Channel offset (digitiser units), one per acquired digital channel	int
60+12N	4*N	Receiver amplitude (linear units), one per acquired receiver channel	int
60+16N	4*N	Receiver SNR (dB), one per acquired receiver channel	float
60+20N	4*K	Cross channel NSR (dB), one per analysed receiver channel pair	float
60+20N+4K	4*N	Sea scatter relative power (dB), one per acquired receiver channel	float

Figura 5: Data

2.4. Base de Datos Madrigal

Es una BD de medidas realizadas desde tierra y de modelos de la alta atmósfera y la ionosfera. Fue desarrollado por el MIT Haystack Observatory en 1980. Estrechamente relacionado

al programa CEDAR (Impulsa la colaboración de esfuerzos en observaciones combinadas). Esta comunidad alberga cerca de 800 investigadores y estudiantes provenientes de todo el mundo.

El elemento central de la base de datos es un formato de datos estándar (NCAR). Capaz de administrar y visualizar datos archivados desde una variedad de instrumentos e instituciones. El propio sitio almacena solo sus propios datos. Los usuarios web, sobre cualquier sitio Madrigal, pueden buscar datos de cualquier otro sitio Madrigal.

3. RESULTADOS

Desarrollamos un script en lenguaje Python con la estructura de Signal Chain que almacenamos con el nombre “bltr_test” que almacenamos en nuestro directorio.

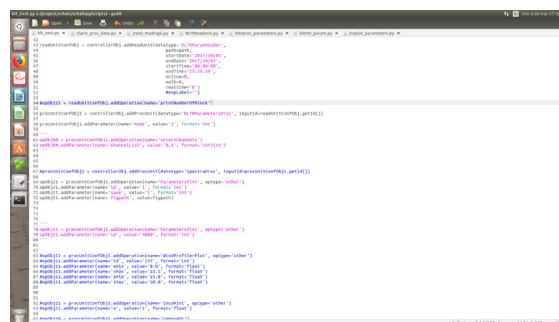


Figura 6: Script - Unidades de lectura, procesamiento y ploteo

4. PLOTEO DE DATOS

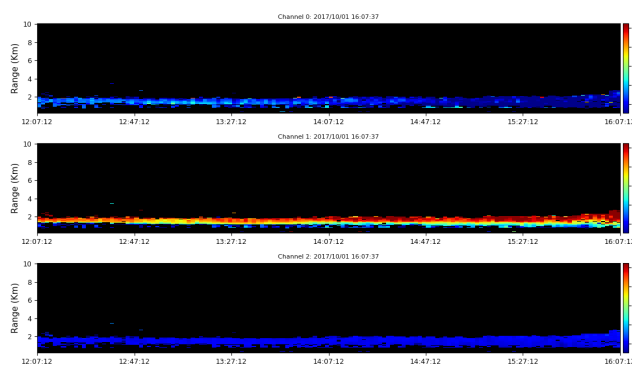


Figura 7 : Datos 01/10/2017 de 12:07 - 16:07

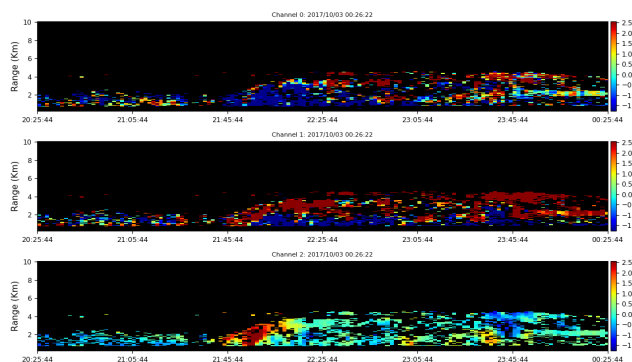


Figura 8 : Datos desde 02/10/2017 20:25 - 03/10/2017 00:25

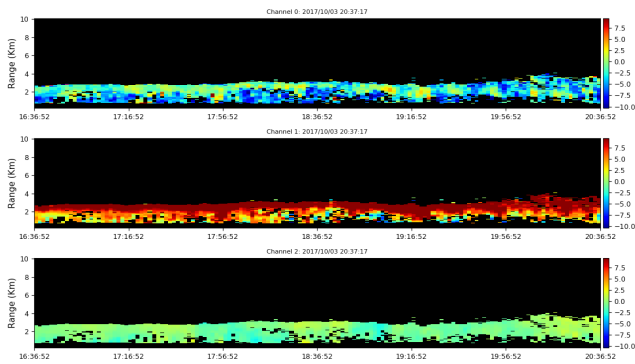


Figura 9 : Datos desde 03/10/2017 16:36 - 20:36

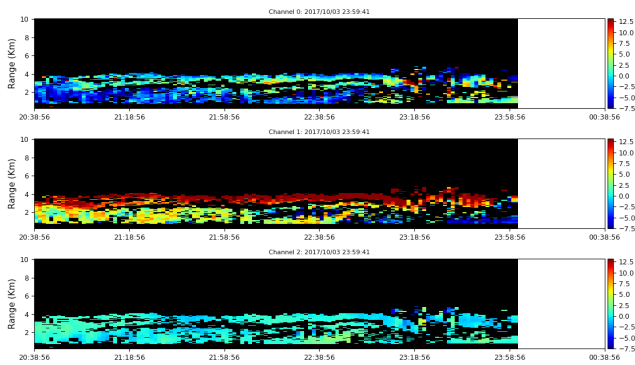


Figura 10: Ploteo de datos desde 03/10/2017 20:38 - 23:59

5. CONCLUSIONES

Para la realización del proyecto se tuvieron que analizar el formato en que los archivos son almacenados en la computadora del radar, esto para conocer su estructura y saber cómo están definidos los parámetros que se miden para así poder efectuar el procesamiento que deseamos.

De los datos tomados de los tres días del BLTR en Huancayo podemos concluir que en el canal 1, que corresponde a la velocidad zonal, es donde se presenta mayor intensidad en la velocidad de los vientos y esto se puede apreciar al ver el color rojo intenso en las figuras de ploteo.

De los datos tomados en el día 2 de octubre especialmente no se encontraron muchas diferencias en las intensidades de los vientos en los 3 canales, pero se mantenía que la mayor intensidad se alojaba en el canal 1.

Concluimos también que en los tres días las velocidades verticales se mantuvieron bastante uniformes en el transcurso de las horas, con excepción entre las 20:25 horas del día 02 de octubre y las 00:25 del día 03 de octubre que es donde vemos en el canal 1 una amplia variedad de colores que corresponden a intensidades de la velocidad de los vientos muy diferentes.

6. RECOMENDACIONES

Para poder añadir una función de monitoreo en tiempo real deberíamos acceder al manual del radar o a su hoja técnica para saber cómo se da el mecanismo de adquisición de datos para así poder leer los archivos por bloques y no esperar al final el día para trabajar con archivos más pesados.

Conectar un disco duro compartido al CPU de la computadora del radar para poder instalar las librerías actualizadas de signal chain y la base de datos madrigal, ya que por la antigüedad del radar esto no es posible realizarlo en la computadora donde actualmente se almacenan los archivos SSWMA.

Instalar la librería Signal Chain en un entorno virtual, dado que es más sencillo y ocasiona menos inconvenientes.

Instalar la base de datos de Madrigal en Debian.

Añadir a la interfaz gráfica de usuario de Signal Chain los botones para trabajar con datos tipo parámetro con los que trabaja el BLTR.

7. AGRADECIMIENTOS

Al Instituto Geofísico del Perú por brindarme las facilidades para realizar el proyecto y al radio observatorio

Al Ing. Juan Carlos Espinoza, por estar a cargo de mi capacitación en el Radio Observatorio de Jicamarca.

Al Ing. Alexander Valdez por su apoyo y predisposición a ayudarme a resolver mis dudas a lo largo del desarrollo del proyecto.

8. BIBLIOGRAFÍA

Signal Chain – Manual del desarrollador: http://jro-dev.igp.gob.pe:3000/projects/signal-chain/wiki/Manual_de_Developador

Signal Chain – Manual de usuario: “http://jro-dev.igp.gob.pe:3000/projects/signal-chain/wiki/Manual_de_Usuario”

SSWMA Data Format – Verson 2. Genesys Software

Madrigal en Jicamarca: Base de Datos Científico “http://intranet.igp.gob.pe/academicos/pdf/ECl/murco_eci2010.pdf”

Python 2.7.16 Documentation, “<https://docs.python.org/2.7/>”

MÉTODOS INVERSOS EN TOMOGRAFÍA DE LA IONÓSFERA ECUATORIAL

* *Inverse Methods in Equatorial Ionospheric Tomography*

Juan Pablo Velásquez Ormaeche

Pontificia Universidad Católica del Perú
Facultad de Ciencias
juan.velasquez@jro.igp.gob.pe

Asesor: Marco Milla

Radio Observatorio de Jicamarca - IGP

RESUMEN

Se describe la implementación de métodos tomográficos de reconstrucción de estructuras ionosféricas de escalas kilométricas, como la llamada Spread-F. Este fenómeno es propio de la ionósfera ecuatorial y consta de regiones donde la densidad volumétrica de electrones decrece abruptamente en comparación a sus vecindades. Con este fin, se realizó una simulación de la ionósfera ecuatorial basada en funciones de Chapman y 3 irregularidades descritas por funciones gaussianas elípticas. Estos perfiles de densidades nos permiten simular data del contenido total de electrones (TEC) de cada trayectoria de las señales Satélite-Receptor dadas por la geometría del sistema. Con esta serie de datos se puede formar un sistema lineal determinado por la matriz de proyecciones, la densidad de electrones que se busca reconstruir y la data TEC simulada. Hay diversos métodos de inversión para reconstruir las densidades de electrones. Los métodos implementados se dividen en 3 categorías: métodos algebraicos (ART, MART y SIRT), regularización de Tikhonov de Orden Cero. Se mostrarán los perfiles reconstruidos aplicando cada método y se evaluará de manera cualitativa las ventajas y desventajas de cada algoritmo.

Palabras clave: [tomografía, spread-f, ionósfera, métodos de inversión]

ABSTRACT

The implementation of reconstruction tomography methods to study ionospheric structures of kilometeric scales, such as the so-called Spread-F, are described. This phenomenon is common to equatorial ionosphere and it consists of regions where the electron density decreases abruptly, in comparison to the neighbor regions. In order to do this, a Chapman-based simulation for the stable atmosphere is performed. 3 elliptical gaussian irregularities are added. These density profiles allow us to simulate total electron content (TEC) data for each Satellite-Receptor trajectory, given by the system's geometry. A linear system is characterized by the projection matrix, the TEC data set, and the reconstructed electron density profiles. There are plenty of inversion methods to reconstruct this electron densities. The implemented methods are divided in 3 categories: algebraic methods (ART, MART and SIRT), 0-th Order Tikhonov regularization. The reconstructed profiles are presented and their performance are evaluated quantitatively.

Keywords: [Ionosphere, Tomography, spread-f, Inversion Methods]

1. INTRODUCCIÓN

1.1. Reseña

La tomografía de la Ionósfera es una técnica mediante la cuál se utilizan señales de satélite para conocer la cantidad de electrones contenidos en su trayectoria hacia el receptor (total electron content, TEC). Con esta información se proveniente de varios satélites y receptores, se puede estimar la densidad volumétrica de electrones en una región de la ionósfera.

1.2. La Ionósfera

La Ionósfera es la capa superior de la atmósfera comprendida entre los 50 km y los 100 km. Se caracteriza por estar parcialmente ionizada (~0.1%) debido a que la radiación Ultravioleta Extrema (EUV) y rayos X, provenientes del sol, liberan electrones de las moléculas de las especies neutras principales (N₂, O y O₂) mediante foto-ionización, formando una combinación de iones y electrones libres (plasma).

1.3. Spread-F Ecuatorial

La dinámica de la ionósfera ecuatorial después del atardecer da lugar a la aparición de estructuras dinámicas como la llamada Spread-F Ecuatorial (Equatorial Spread-F o burbujas de plasma). Consiste en regiones donde la densidad de electrones por metro cúbico disminuye drásticamente en comparación a las regiones vecinas.

Las abruptas gradientes en los bordes de las burbujas de plasma ocasionan la degradación de las señales de comunicación satelital y de navegación.

2. DESARROLLO

2.1. Modelo

Se utilizó el modelo de Chapman, de una ionósfera en equilibrio térmico e hidrostático, con 3 irregularidades gaussianas elípticas, como las mostradas en la figura 1.

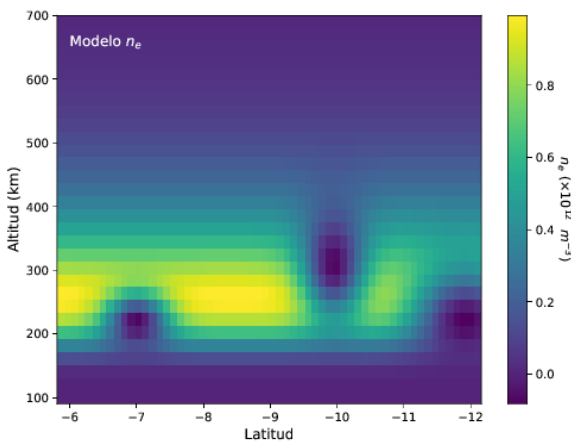


Figura 1. Modelo de una Ionósfera de Chapman con irregularidades gaussianas elípticas.

2.2. Geometría

Se dispuso de 7 receptores y 80 posiciones de satélite y se dividió la ionósfera en una grilla polar.

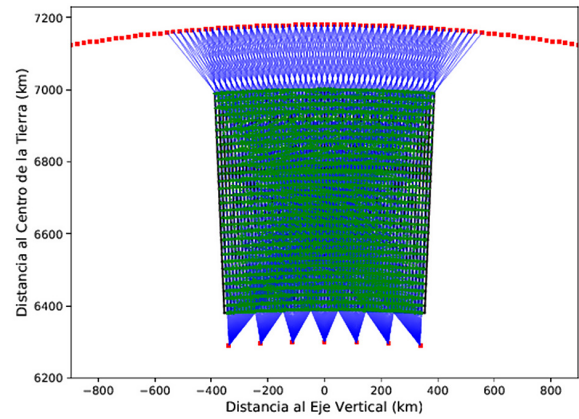


Figura 2. Grilla de la Ionósfera de 60x30. 7 receptores y 80 posiciones de satélite.

2.3. Construcción del Sistema Lineal

La carga total contenida (TEC) en la trayectoria de una señal emitida del satélite al receptor es descrita por:

$$TEC = \int_s n_e ds(1)$$

Esta ecuación se puede discretizar y expresar, mediante el uso de la grilla mencionada, de la forma:

$$TEC \approx \sum_a^b n_e \Delta s$$

Para una configuración de diversos receptores y posiciones de satélite, se puede formar un vector **b** de mediciones de TEC, de manera que la ecuación nos lleva al sistema lineal:

$$b = \hat{A}n_e + \epsilon$$

Donde \hat{A} es una matriz que contiene la información de la geometría del sistema: contiene la longitud del segmento que cada trayectoria que cruza el grid en cada celda. ϵ es el ruido asociado a cada medición de TEC.

El problema que la tomografía busca resolver es obtener la densidad volumétrica de electrones n_e de la información provista por \hat{A} y **b**. Con este fin, es que constantemente se implementan nuevas técnicas de inversión para lograr este cometido. El núcleo de este trabajo está basado en la implementación de diversas técnicas de inversión (con distintos enfoques) y en el estudio de su eficiencia.

3. ALGORITMOS DE RECONSTRUCCIÓN

3.1. Algoritmos Algebraicos

3.1.1. ART (Additive Algebraic Technique)

Converge a la solución de mínimos cuadrados. Consiste en proyectar una solución estimada inicial sobre el hiper espacio producido por la primera fila de \hat{A} , luego, esto se proyecta sobre el hiper espacio de la segunda fila y así sobre todas las filas hasta alcanzar la convergencia.

$$n_e^{(k+1)} = n_e^{(k)} + \lambda_k \frac{\overbrace{b_i - \langle \hat{A}^i, n_e^{(k)} \rangle}^{\text{medicin-prediccion}}}{\underbrace{\langle \hat{A}^i, \hat{A}^i \rangle}_{\text{tamaño rayo en celda}}} \hat{A}^i$$

La elección del parámetro de relajación λ_k adecuado ayuda a acelerar proceso de convergencia.

3.1.2. MART (Multiplicative Algebraic Reconstruction Technique)

Converge a la solución de máxima entropía (información). Toma en cuenta la positividad de la solución

$$(n_e^k \geq 0).$$

$$n_e^{(k+1)} = n_e^{(k)} \left(\frac{b_i}{\langle \hat{A}^i, n_e^{(k)} \rangle} \right)^{\lambda_k \frac{\overbrace{\hat{A}^i}^{\text{tamaño rayo en celda}}}{\underbrace{\langle \hat{A}^i, \hat{A}^i \rangle}_{\text{mediccion prediccion}}}}$$

3.2. Regularización de Tikhonov

Dado el hecho de que la matriz es singular, la idea principal de los métodos de regularización es de agregar restricciones de carácter físico al proceso de minimización del funcional que describe la discrepancia entre los datos TEC “reales” y el TEC simulado. En lugar de minimizar el funcional:

$$\|\hat{A}x - b\|_2^2 = 0$$

Donde x es la densidad de electrones. Se minimiza una función de costo que contenga restricciones físicas. La forma general de describir esta función de costo es:

$$J(\mathbf{x}) = \|\hat{A}x - b\|_2^2 + \lambda^2 \|L(x - x_0)\|_2^2$$

Donde L es la matriz que impone las restricciones físicas y/o matemáticas de la regularización, siendo el caso más básico el de orden cero: L = identidad. L de mayores órdenes incluyen matrices de derivadas de primer y/o segundo orden. Este estudio sólo comprenderá el caso de orden cero. El llamado parámetro de regularización es λ y, su importancia, se explicará en la siguiente subsección.

3.3. Selección del Parámetro de Regularización:

Podemos ver en nuestra función de costo:

$$J(\mathbf{x}) = \|\hat{A}x - b\|_2^2 + \lambda^2 \|x\|_2^2$$

El parámetro de regularización provee un balance entre la discrepancia con la data y las restricciones añadidas al funcional. Como ya se mencionó, estas restricciones pueden tener diferentes formas. Estudiaremos el caso más simple. Se utilizaron dos métodos para esta selección: el criterio de la Curva-L y Validación Cruzada Generalizada

3.3.1. El Criterio de la Curva-L:

Consiste en dibujar un gráfico logarítmico de la discrepancia con la norma y la restricción añadida el funcional de costo. Exhibe una forma característica de L y el parámetro de regularización óptimo se encuentra en la esquina de la curva:

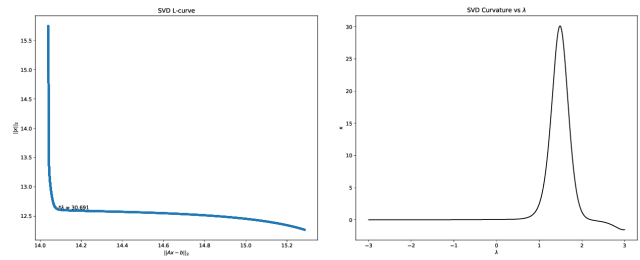


Figura 3a: Curva-L para 100 receptores, 5 satélites y 4% de ruido. A la derecha (3b): la función de curvatura respecto a λ .

Utilizamos la expresión de curvatura para curvas paramétricas:

$$\kappa = \frac{x'y'' - y'x''}{((x')^2 + (y')^2)^{\frac{3}{2}}}$$

3.3.2. Validación Cruzada Generalizada:

Es una práctica común en problemas de Inversión el reservar una porción de la data disponible como “training data” para calcular el modelo estimado. Este mismo es usado para predecir la data restante y, por consiguiente, estimar el parámetro de regularización adecuado. De ahí a que su nombre sea Validación Cruzada.

Dos desventajas son palpables: Al resgrentir la cantidad de data a usar, el modelo estimado pierde información respecto al obtenido con toda la data. No existe un criterio claro que nos permita decidir cuánta data separar para realizar la validación.

$$V_0 = m \frac{\|\hat{A}x - b\|_2^2}{Tr(I - \hat{A}A^\dagger)}$$

La función a minimizar es V_0 con respecto a λ .

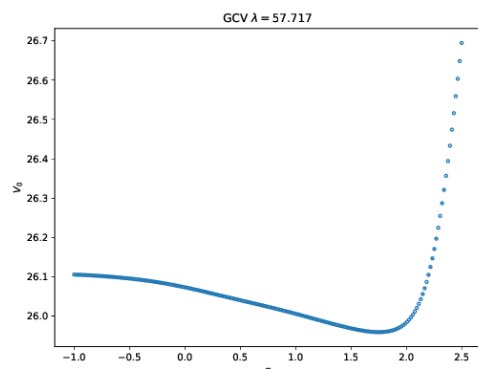


Figura 4: V_0 para 100 receptores, 5 satélites y 4% de ruido. No siempre muestra un mínimo global bien definido.

4. RESULTADOS

En esta sección, se mostrarán los diferentes resultados de cada método de reconstrucción. Estudiamos la eficiencia de reconstrucción de cada algoritmo y cómo esta varía con el ruido de las mediciones y el número de receptores.

4.1. Error Porcentual vs. Ruido TEC:

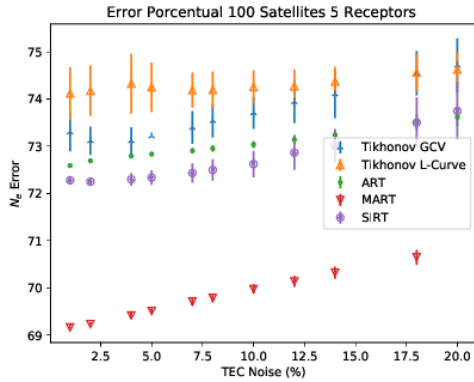


Figura 5: Error porcentual para 100 posiciones de satélite y 5 receptores.

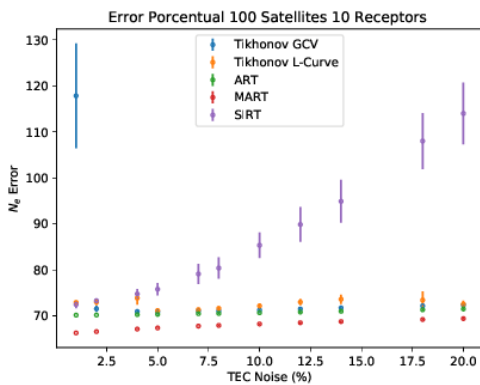


Figura 6: Error porcentual para 100 posiciones de satélite y 10 receptores.

4.2. Error Porcentual vs. Número de Receptores:

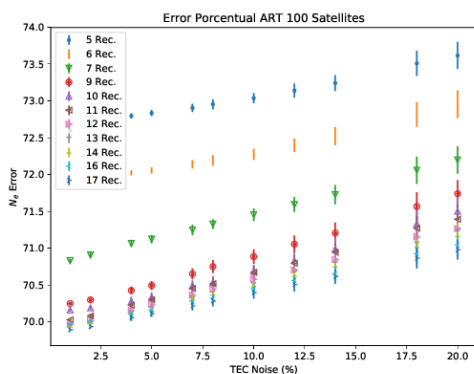


Figura 7: Error porcentual de ART respecto al ruido TEC y el número de receptores.

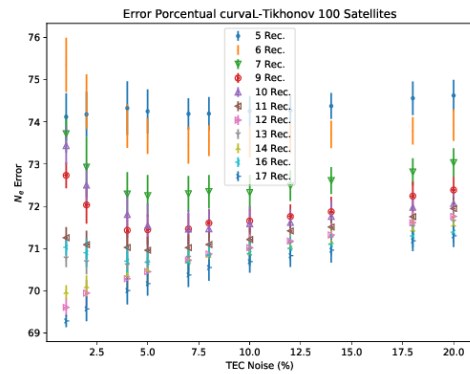


Figura 8: Error porcentual de Tikhonov vía la curva-L respecto al ruido TEC y el número de receptores.

5. DISCUSIÓN

Los métodos algebraicos son sencillos y no requieren de mucho poder computacional para producir resultados aceptables, pero no tienen ningún tipo de restricciones físicas, como la positividad de la solución en MART (que es una consecuencia algebraica mas no una imposición física) y no en ART ni en SIRT.

La introducción de funciones de costo al proceso de minimización nos permite imponer restricciones físicas a las soluciones, pero introduce una nueva dificultad: encontrar el parámetro de regularización óptimo λ (como en la ecuación (ec. Tikhonov)), que pesa la función de costo con la discrepancia entre los datos simulados y los reconstruidos.

Técnicas como la inspección de la curva L son recomendables para problemas que tengan un solo parámetro de regularización. Una curva L se construye dibujando la relación entre la norma de la discrepancia entre la simulación y la reconstrucción, y la norma de la solución. Esta gráfica suele tener una forma de L y el valor óptimo del parámetro de regularización es el que produce la esquina de la gráfica.

El método de la Curva L es sencillo, ya que sólo se necesita realizar una inspección visual para estimarlo (mejor aún si usamos la expresión de la curvatura mostrada en su respectiva sección, pero no es práctico para funcionales de costo con más de un parámetro regularización. El método de la Validación Cruzada Generalizada también es sencillo de implementar y nos ofrece una manera de corroborar los resultados obtenidos mediante el método de la Curva-L.

6. CONCLUSIONES

Las reconstrucciones obtenidas por los métodos algebraicos son menos sofisticadas, pero producen resultados relativamente confiables, pero fallan al reproducir estructuras ionosféricas más complejas como la mostrada en la figura 1. Si bien los métodos de regularización de Tikhonov incluyen restricciones en el espacio de soluciones, estas parecen ser muy rudimentarias para reconstruir estructuras más complejas.

La selección del parámetro de regularización adecuado sigue siendo uno de los aspectos cruciales en este análisis. La Validación Cruzada Generalizada arroja resultados más eficientes respecto a lo de la curva-L, la mayoría de las veces, pero se muestra inesperado para los casos de 1% de ruido para 7, 10 y 13 receptores.

Se puede apreciar que MART funciona mejor para todos los casos y que el error de la reconstrucción disminuye a medida que aumenta el número de receptores.

Los algoritmos reconstruyen pobremente las características morfológicas de las irregularidades.

7. AGRADECIMIENTOS

Al Instituto Geofísico del Perú a través del Radio Observatorio de Jicamarca por darme la oportunidad de desarrollarme profesionalmente y apoyarme a realizar mi tesis.

Al Dr. Marco Milla, por sus claras directivas y apoyo constante. A Enrique Rojas y Gonzalo Cucho, por sus observaciones y consejos. A Frank Meléndez y Joseph Prado, por hacer el día a día en la oficina más ameno e interesante.

A mi padre, Carlos Manuel Velásquez Rondón, por todo.

8. BIBLIOGRAFÍA

Kelley, M. C., The Earth's Ionosphere. Academic Press, San Diego, 1989.

Richard C. Aster, Brian Borchers, and Clifford H. Thurber. Parameter Estimation and Inverse Problems. Elsevier Academic Press, 2005.

Comberiate, J. M., F. Kamalabadi, and L. Paxton (2006), Tomographic imaging of equatorial plasma bubbles, Geo-phys. Res. Lett., 33, L15805,.

MONITOREO DEL ESPECTRO DE POTENCIA PARA DETECCIÓN DE BEACONS SATELITALES

** Monitoring the cross-power spectrum for detection of satellite beacons*

Jhoseling Milagros Melgarejo Figuerao

Universidad Nacional de Ingeniería
Facultad de Ciencias
jhoseling.melgarejo@jro.igp.gob.pe

Asesor: Dr. Edgardo Pacheco

Radio Observatorio de Jicamarca - IGP

RESUMEN

En el presente trabajo se expone la revisión de la implementación del procesamiento de dos señales de radiofrecuencia en las bandas UHF y VHF emitidas por transmisores satelitales, en primera instancia, el TEC relativo usando la técnica Doppler Diferencial. Así mismo, se muestran los resultados de detección de la variación de TEC utilizando las señales de radio frecuencia transmitida por satélites que se encuentran operativos como COSMOS, CASSIOPE y DMSP-F15

Palabras clave: radiofrecuencia, satélites, contenido total de electrones

ABSTRACT

In the present work, a review of the implementation of the processing of two radiofrequency signals in the UHF and VHF bands emitted by satellite transmitters is presented, in the first instance, the relative TEC using the Differential Doppler technique. Likewise, the detection results of the TEC variation will be shown using the radio frequency signals transmitted by operational satellites such as COSMOS, CASSIOPE, and DMSP-F15

Keywords: radiofrequency, satellites, total electron content

1. INTRODUCCIÓN

El instrumento transmisor conocido como "radio beacon" se utiliza como carga útil de nanosatélites para emitir señales de radiofrecuencia RF y que se propaguen atravesando la ionósfera. Para recibir las señales de satélites se usan estaciones receptoras terrenas. Las señales que llegan del receptor viajando desde los satélites que se encuentran aproximadamente a 1,000 Km de altura y suelen tener transmisores en bandas VHF y UHF las cuales son más sensibles a los fenómenos ionosféricos. Cuando el satélite se encuentra orbitando en el espacio sobre la estación receptora las señales captadas se ven afectadas por una diferencia de fases entre estas. Esto permite a través de una técnica de medición el cálculo de nuestro parámetro de interés que es el contenido total de electrones relativo (rTEC) en la ionósfera. Dichas mediciones obtenidas permitirán investigar la ionósfera ecuatorial y observar eventos de interés como cuando se presenten irregularidades. Los datos resultantes son ampliamente usados en la comunidad científica para estudios locales y globales del comportamiento de la ionósfera.

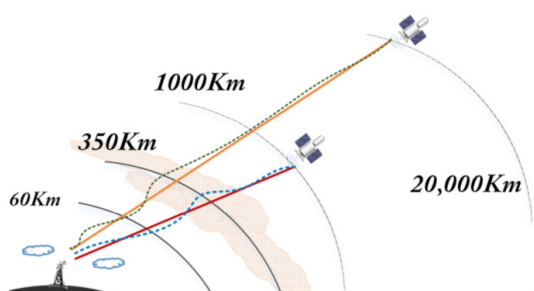


Figura 1: representación gráfica de la ubicación de los satélites captados

La estación receptora terrena permite recibir las señales de transmisores a bordo de satélites de baja órbita o LEO (Low Earth Orbit) que se propagan a través de la ionósfera para posteriormente realizar el procesamiento de estas señales y estimar la variación del TEC cuando se produzca el enlace directo entre el satélite y el receptor. Las señales de radiofrecuencia del transmisor o radio beacon captadas corresponden a las bandas VHF y UHF (150 MHz y 400 MHz) y son transmitidas en fase.

2. PROCESAMIENTO DE SEÑALES

2.1. Calculo del contenido total de electrones

Se define el contenido total de electrones entre dos puntos A y B, como el número total de electrones libres contenidos en una columna de un m que une A con B, es decir:

Con estas piezas en su totalidad, se ensambla y se tiene toda la estructura, para poder obtener el movimiento vertical, que se necesita

$$TEC = \int_B^A Ne \cdot dz$$

Los valores para TEC suelen variar entre $(10^{19}$ y 10^{26} [e/m^2]) dependiendo:

- de la hora del día
- la estación del año
- la actividad solar

- la actividad geomagnética
- la latitud magnética

La técnica más común para calcular el TEC es a partir de dos observables. Estos observables son parámetros que se derivan de las mediciones de GPS, los cuales son en base al tiempo y las diferencias de fase basadas en la comparación entre la señal procedente del satélite y la réplica de dicha señal generada por el receptor.

2.2. Preprocesamiento de señales

En esta sección se procederá a realizar una descripción del procesamiento de las señales adquiridas, datos adquiridos y reprocesados de las señales de 150 y 400 MHz son almacenados como RAWDATA o datos crudos. Los datos llegan con una tasa de muestreo de 1.25 Msps por cada canal, lo cual debido a la digitalización en cuadratura da un ancho de banda de igual valor. Esto significa que si la frecuencia central era 400 MHz, los límites inferiores y superiores son 399.375 MHz y 400.625 MHz, asimismo con 150 MHz. Esto permite poder capturar todos los satélites que transmiten este par de frecuencias.

El objetivo del preprocesamiento es reducir el peso del archivo al concentrar la información en la banda deseada. Asimismo se limpia la señal de interferencias lejanas a la banda de interés. De este modo se puede validar si se ha capturado la señal o si sólo se ha grabado señales indeseadas.

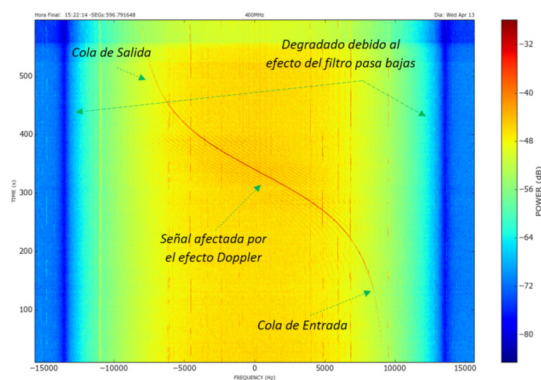


Figura 2: espectrograma del archivo reprocesado proveniente de 150MHz [5]

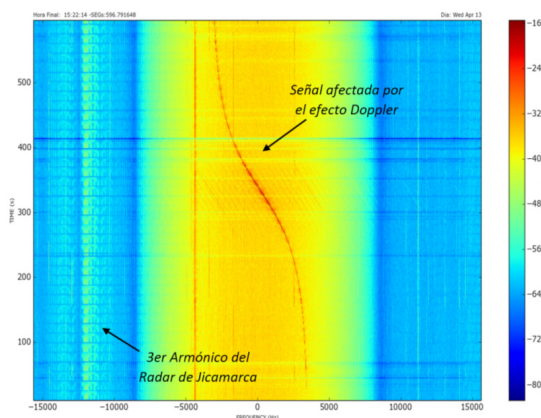


Figura 3: espectrograma del archivo reprocesado proveniente de 400MHz [5]

En la figura 3 se puede ver el espectrograma resultante, se modifica el valor de potencia para que el mínimo valor sea 0 dB. Se puede ver que los niveles de las interferencias decaen y los niveles de la señal deseada aumentan. Estos niveles son utilizados

para formar la función arcotangente que acusa el efecto Doppler.

Los datos procesados serán subidos en un servidor local del framework Django en la versión 2.11 para que puedan ser visualizados se sigue el siguiente diagrama de flujo del programa en Django.

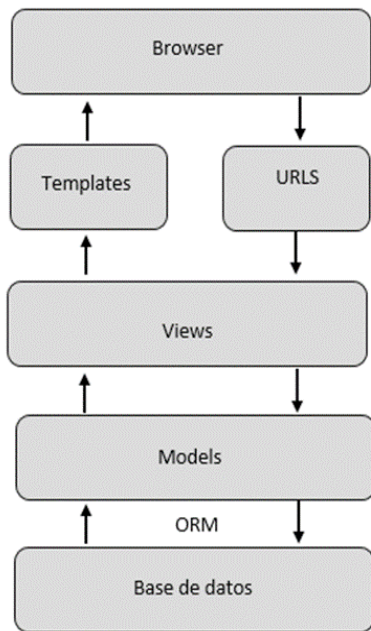


Figura 4: el diagrama de flujo del programa en Django.

3. RESULTADOS

El preprocesamiento de las señales nos genera carpetas de la data cruda, la data decimada y el pase de los satélites cada una agrupada en una carpeta por días los datos trabajados son de la fecha 02-01-2018.

El código en Python realizado en ruta a la carpeta (02-01-2018) que contiene los archivos mostrados anteriormente, estos son procesados por el programa final_processCedar.py

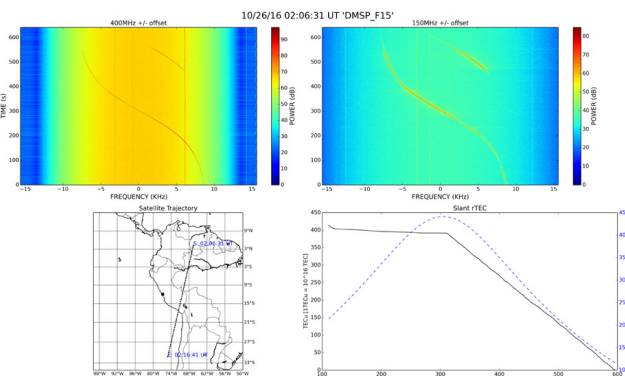


Figura 5: imagen superior espectrograma del archivo procesado proveniente de 150MHz - 400MHz, imagen inferior izquierda trayectoria del pase satelital imagen inferior derecha Grafica rTEC a través del tiempo para la fecha de 10/26/16

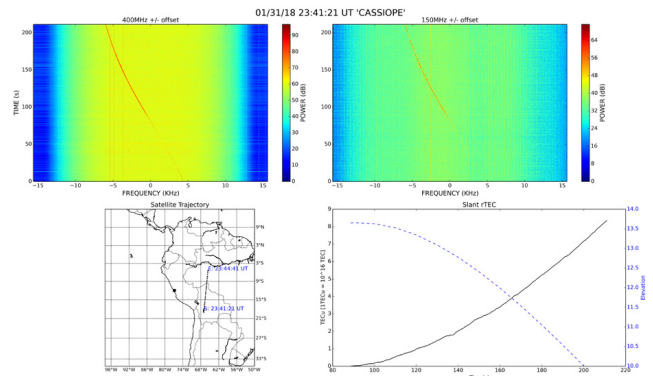


Figura 6: imagen superior espectrograma del archivo procesado proveniente de 150MHz - 400MHz, imagen inferior izquierda trayectoria del pase satelital imagen inferior derecha Grafica rTEC a través del tiempo

4. DISCUSIÓN

- Se puede apreciar cómo es el efecto Doppler desvía a la señal unos tantos KHz, pero es menor para frecuencia de 150 MHz que para la señal de 400 MHz. Asimismo, se puede apreciar cómo en las colas la señal se degrada, pero no tanto como la señal de 400 MHz, esto puede ser debido a diversos factores, como la sensibilidad de la antena o por la pérdida de potencia debido a la distancia.

- El efecto Doppler se puede observar en la figura 4 donde la desviación de frecuencia de las señales de beacon va desde los -3 KHz hasta los 3 KHz para la señal de 150 MHz y de -9 a 9 KHz para la señal de 400 MHz se puede apreciar la interferencia del 3er armónico del radar de Jicamarca muy cerca de la señal. Por eso se debe tener cuidado y tener en consideración los momentos en que opera el radar del ROJ. Con el proceso de decimación y filtrado descrito se evitan esas interferencias.

- La resolución de los espectrogramas dependerá del nivel de muestras que ingresen a la FFT (Fast Fourier Transform), y del nivel de muestras que se traslapan y del número de espectros que se integran.

- De los datos del satélite se puede visualizar una anomalía la cual se ve entre los 600 y 650 segundos. Así mismo se puede ver que los niveles de TECus han decrecido por encontrarse de noche, en donde el nivel de electrones desciende.

5. CONCLUSIONES

Se desarrolló este trabajo con el objetivo de poder iniciar el proceso de optimización de algoritmos utilizados en la adquisición y procesamiento de datos TEC.

Un satélite se puede ubicar a través de la estimación de su órbita, a través de los parámetros de las efemérides o TLE's (Two Line Elements) que se generan a partir de los datos que envían.

Se utiliza un archivo de texto llamado satellites.txt el cual contiene los nombres y offsets de los satélites de interés. Estos nombres son buscados en las líneas de los nombres de satélites en la información descargada. En caso de encontrar el nombre del satélite de interés, la información de sus efemérides es almacenada.

Cuando un pase de satélite cumple estos criterios, se guarda todos los datos entregados por la función antes mencionada en un archivo con extensión *.pass. Estos datos sirven más adelante para poder localizar y referenciar las medidas del TEC de dichos pases.

Los pases de satélite que se encuentren en la misma ventana de tiempo se juntan uniendo los tiempos de inicio y fin según sea conveniente para poder grabar más de una señal en caso pasen varios satélites al mismo tiempo.

Se utilizó el framework Django para generar un entorno de visualización local del procesamiento de señales.

6. RECOMENDACIONES

Se recomienda preinstalar los siguientes módulos:

Para poder obtener el rTEC se necesitan datos grabados con una frecuencia de muestreo de 31 250 sps(samples per second) y frecuencia central en la frecuencia del beacon satelital.

De igual manera los bloques de datos es adquirido con la misma frecuencia de muestreo, pero con frecuencia central en el 2do beacon, beacon 1 = 150.012 MHz beacon 2 = 400.032MHz

Se requiere de un archivo *.pass el cual contiene información del pase satelital, como tiempo y trayectoria.

Opcionalmente se puede contar con los resultados de LISN en una carpeta.

Trabajar con las librerías Python para la obtención de las gráficas.

Librerías Python utilizadas

- Import sys
- Import datetime
- Import os
- import numpy as np
- Import h5py
- Import math
- From matplotlib.pyplot import *
- Grom mpl_toolkits.basemap import Basemap
- From scipy import *
- Import optparse
- Import time
- Import glob

7. AGRADECIMIENTOS

Mi agradecimiento al Instituto Geofísico del Perú a través del Radio Observatorio de Jicamarca, al Dr. Edgardo Pacheco por su apoyo y a mis compañeros del programa por los gratos momentos compartidos.

8. BIBLIOGRAFÍA

[1]Wikipedia, "Two-line element set." [Online]. Available: https://en.wikipedia.org/wiki/Two-line_element_set

[2]E. Research, "n200/n210 product overview," pp. 1–2. [Online]. Available: https://www.ettus.com/content/files/07495_Ettus_N200-210_DS_Flyer_HR_1.pdf

[3]J. Vierinen, "Beacon satellite receiver for ionospheric tomography," p. 2, 2013.

[4]E. Kudeki, Applications of Radiowave Propagation. University of Illinois at UrbanaChampaign.

[5] "Diseño e implementación de un sistema del contenido total de electrones (TEC) relativo", Josemaría Gómez Socola.

OPTIMIZACIÓN DEL TELESCOPIO NISHIMURA DEL INSTITUTO GEOFÍSICO DEL PERÚ PARA SU USO CIENTÍFICO EN ASTRONOMÍA

**Optimization of the Geophysical Institute of Peru's Nishimura telescope for scientific use in astronomy*

Mario Zegarra-Valles

Universidad Nacional de Ingeniería
Facultad de Ciencias
Escuela Profesional de Física
mario.zegarra.valles@gmail.com

Asesor: Dr. Nobar Baella Pajuelo

Unidad Funcional de Astronomía Observatorio de
Huancayo- IGP

RESUMEN

Este trabajo tiene como objetivo optimizar el telescopio Nishimura (NSH60) del Instituto Geofísico del Perú para uso científico en astronomía. Se han realizado investigaciones sobre las principales características del NSH60 y también sobre las técnicas de optimización de telescopios. Como resultado de nuestro estudio, se ha desarrollado un software de enfoque básico. Además, se construyó una interfaz para el autoguiado del NSH60. También, se escribió un controlador especial para ampliar las capacidades de control del instrumento. Todas estas mejoras están implementadas actualmente en el telescopio. Finalmente, describimos las observaciones de ciertos eventos astronómicos llevados a cabo durante nuestro trabajo de optimización.

Palabras clave: Telescopio Nishimura, enfoque, autoguiado, ASCOM.

ABSTRACT

This work aims to optimize the Geophysical Institute of Peru's Nishimura telescope (NSH60) for scientific use in astronomy. Research has been carried out on the main characteristics of the NSH60 and also on the telescopes optimization techniques. As results of our study a basic focusing software has been developed. In addition, an interface for the autoguiding of the NSH60 was built. A special driver was also written to expand the control capabilities of the instrument. All these improvements are currently implemented in the telescope. Finally, we outline the observations of certain astronomical events carried out during our optimization work.

Keywords: Nishimura telescope, focusing, autoguiding, ASCOM.

1. INTRODUCCIÓN

En la ciudad universitaria de la *Universidad Nacional "San Luis Gonzaga" de Ica*, exactamente en el lugar antiguamente conocido como *Estación Solar de Ica*¹, está instalado un telescopio reflector Cassegrain clásico de 600 mm de apertura de marca Nishimura (NSH60). Este instrumento está asentado sobre una montura ecuatorial de tipo horquilla (Figura N° 1), con un controlador, equipos de manejo anexos y accesorios. El NSH60 pertenece al *Instituto Geofísico del Perú*. Fue donado por ciudadanos del Japón en reconocimiento a las labores científicas y de difusión astronómica de uno de sus selectos miembros: el Dr. Mutsumi Ishitsuka Komaki (1930-2018) [5]. Desde marzo hasta noviembre (salvo julio) del 2017 y en mayo y junio del 2018, una vez cada mes, por dos noches consecutivas en cada caso, se realizaron pruebas con el telescopio. El objetivo fundamental de éstas fue mejorar nuestro conocimiento del NSH60 (la empresa fabricante no proporcionó un manual técnico detallado del uso del mismo). Aún existen ciertas cuestiones tecnológicas a resolver antes de emplear el NSH60 en observaciones astronómicas con consistencia científica (fotometría estelar, por ejemplo). En este trabajo presentamos nuestras primeras actividades y logros orientados a optimizar este instrumento.



Figura 1: El telescopio Nishimura (NSH60).

2. DESARROLLO

2.1. Características generales del NSH60

Las principales características generales del telescopio se muestran en la tabla N° 1 [7, 8 y 9].

2.2. Documentación y Optimización de uso del software de proveedor

Antes de iniciar la optimización se ha redactado un breve manual para operar el telescopio con el debido cuidado. En él, damos las advertencias básicas usuales con relación a este tipo de

¹ Actualmente, es el Centro de Investigación de la Actividad Solar y sus Efectos sobre la Tierra (CIASEST), aunque también posee otro nombre alternativo: Instituto de Investigación Científica de la Actividad Solar "Mutsumi Ishitsuka Komaki" (según una placa situada en la sala de control del telescopio).

Telescopio Nishimura Reflector Cassegrain clásico	
Espejo primario	
Forma	Paraboloide de revolución cóncavo
Material	Vidrio cerámico de bajo coeficiente de dilatación térmica
Recubrimiento	SiO ₂ + Al
Diámetro	600 mm
Espesor (borde)	89,6 mm
Distancia focal	1800 mm
Razón focal	3,00
Captación lumínica	7347x
Magnitud visual estelar límite	16,7
Resolución (Rayleigh)	0,23"
Diámetro del agujero central	150 mm
Diámetro efectivo	580,9 mm
Razón focal efectiva	3,10
Captación lumínica efectiva	6887x
Magnitud visual estelar límite efectiva	16,7
Precisión P-V (superficie)	$\lambda/8$
Precisión RMS (superficie)	$\lambda/40$, a 633 nm
Precisión Strehl	> 0,9
Espejo secundario	
Forma	Hiperboloide de revolución convexo
Diámetro	190 mm
Espesor (borde)	30 mm
Distancia focal	531 mm
Obstrucción (con soporte)	> 10,0 %
Sistema conformado por los espejos primario y secundario	
Distancia focal	5993 mm
Razón focal	10,0
Diámetro efectivo	569,1 mm
Razón focal efectiva	10,5
Captación lumínica efectiva	6610x
Magnitud visual estelar límite efectiva	16,6
Montura ecuatorial, de horquilla y computarizada, con motores de ascensión recta y declinación	

Tabla 1. Algunas características generales del telescopio Nishimura

instrumental óptico, describimos la manipulación de la caseta que protege al telescopio y detallamos algunos de los pormenores sobre el controlador, la unidad de control de los servomotores de ascensión recta y declinación, el controlador manual y el programa *The Master of Telescope* (TMOT) que a la fecha es usado durante programas de divulgación científica para dirigir el NSH60 hacia objetivos celestes (Figura N° 2). En ese sentido se consiguió identificar el mejor adaptador USB-Serial compatible con el programa TMOT para su uso con cualquier computador.

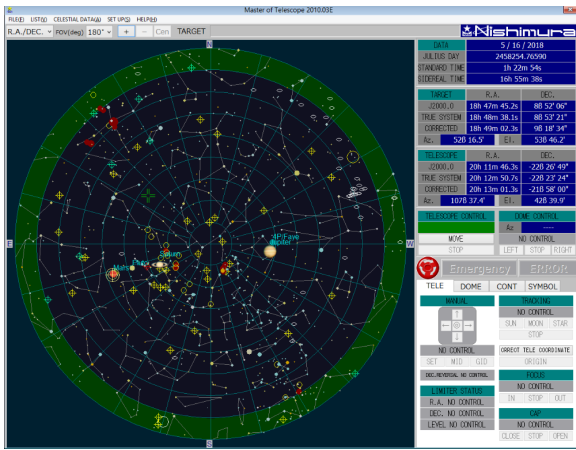


Figura 2: Ventana de The Master of Telescope (TMOT)

2.3. Enfoque del NSH60

Se modeló un enfoque del NSH60 simplificado, esto es, del sistema conformado únicamente por los espejos primario y secundario del telescopio, asumiendo un comportamiento de la luz conforme a las reglas de la óptica geométrica (Figura N° 3), con los rayos luminosos ingresando al instrumento ya sea paralelamente a su eje óptico o en todo caso un tanto inclinados. Esto con la finalidad de visualizar mejor cómo se da el enfoque del telescopio, que ocurre por un desplazamiento, no tan precisamente controlable (Figura N° 4), de su espejo secundario. Además, para calcular la distancia focal comprometida. Grosso modo, se observa que un pequeño desplazamiento del espejo secundario genera un magnificado desplazamiento del foco del equipo. La distancia focal del NSH60 (Tabla N° 1), es de aproximadamente 5993 mm (cf. 6000 mm, valor del fabricante).

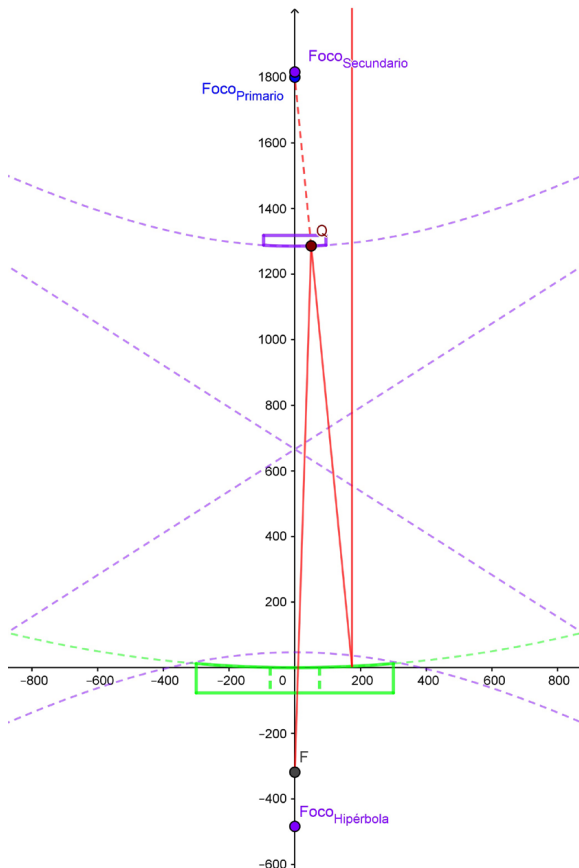


Figura 3: Simulación en GeoGebra [4] del enfoque del NSH60 (unidad: mm).



Figura 4: Júpiter e Io (27/05/2019), usando el NSH60 con una cámara CMOS de marca ZWO, modelo ASI 071 MC, del Observatorio y Planetario Moche. El actual sistema de enfoque del NSH60 aún necesita afinarse con un accesorio especializado ubicable en su porta-ocular.

2.4. Software de enfoque del NSH60

Hemos desarrollado un programa (FocaliS) que asiste al observador astronómico durante la fase del enfoque óptico de la imagen de una estrella producida por el telescopio sobre el *imaging chip* de una antigua cámara CCD de marca SBIG, modelo ST-7 [12], de la *Unidad de Astronomía del Instituto Geofísico del Perú*. Este código fue escrito en C/C++, usando la *Microsoft Windows Application Programming Interface (Win32 API)* [10] integrada en el *Microsoft Visual Studio 6*, aprovechando la existencia de una biblioteca y un *driver* universales de la propia *Santa Barbara Instrument Group (SBIG)* [13]. En general, nuestro software permite lo siguiente:

- Control y monitoreo de temperaturas del entorno inmediatamente próximo a la cámara y del interior de la cápsula que contiene su sensor CCD. Es posible disminuir esta última temperatura para mejorar la razón señal/ruido de las imágenes.
- Control de los parámetros de integración para cualquier adquisición: *binning*, *substracción* (o no) de un *dark frame* y tiempo de exposición.
- Control de enfoque asistido: Se usa un cursor tipo caja para seleccionar una imagen desenfocada de alguna estrella de magnitud adecuada. A continuación, el programa automáticamente se concentra en este astro, capturándolo en sucesivas integraciones acotadas por nuestro cursor. Entonces, en cada una de las imágenes resultantes se calcula el centroide de la estrella y su perfil radial. Finalmente, en este perfil radial se realiza un ajuste según *Levenberg-Marquardt* [6] para las tres curvas teóricas actualmente implementadas: una gaussiana (Figura N° 5), una curva *Moffat* y una curva empleada en el programa *Aperture Photometry Tool* [1], definida por un polinomio quintico acoplado a una gaussiana. El ajuste nos proporciona un parámetro importante: la anchura a la mitad de su altura máxima, esto es, el valor *FWHM (full width at half maximum)*. Nuestro criterio de enfoque consiste en encontrar el mínimo *FWHM* posible para la estrella. Para ello se desplaza lentamente el espejo secundario del telescopio por medio de su controlador manual hasta encontrar el valor mínimo de *FWHM* (en ese momento se dice que el NSH60 está enfocado).

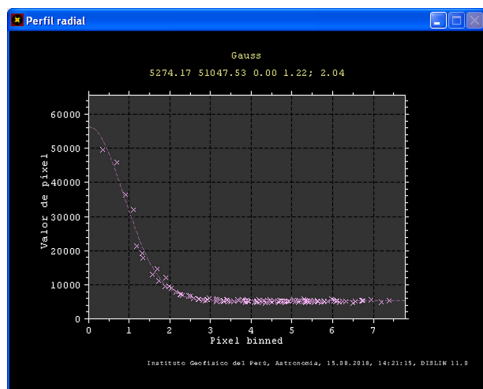


Figura 5: Nuestro programa FocalIS, mostrando un perfil radial estelar y con un ajuste gaussiano.

2.5. Interfaz de Autoguiado del NSH60

Diseñamos y fabricamos una interfaz que sirve de intermediaria entre la cámara CCD SBIG ST-7 y el telescopio. Esta interfaz (Figura N° 6) nos permite usar la cámara como un *autoguider* o *self-guider* para el NSH60. Durante el seguimiento de un astro, existen pequeños errores acumulativos que producen que este *tracking* sideral no sea muy preciso. La aplicación de largos tiempos de integración para el registro de objetos astronómicos de bajo brillo (razón señal/ruido baja) es impracticable sin una debida y permanente corrección al seguimiento de nuestro astro de estudio. Así, justo cuando una estrella previamente centrada con la cámara se sale de dicho centro, el *software* propio de la cámara entra en acción indicándole al telescopio hacia dónde debe moverse para que la estrella no salga del campo de trabajo. Sin embargo, la CCD puede enviar mensajes al NSH60 sólo a través de un circuito traductor (nuestra interfaz) con el fin de mover los motores de ascensión recta y declinación discretamente [11]. Estos movimientos correctivos tienen como resultado la rectificación ininterrumpida del seguimiento estelar durante la correspondiente jornada de observación astronómica.



Figura 6: Interfaz IGP ASTRO NSH60 001, desarrollado en este trabajo y empleado en el autoguiado del telescopio NSH60 con la cámara CCD SBIG ST-7.

2.6. Desarrollo de driver ASCOM exclusivo para el NSH60

A partir de una plantilla C# elaborada por la ASCOM Initiative [2], hemos desarrollado un driver que permite utilizar el telescopio no solamente con su *software* original, sino también con otros como *TheSkyX* o *Cartes du Ciel*. Esto nos proporciona una mayor versatilidad que la que el *software* del proveedor (TMOT) posee, mejorando nuestra capacidad de comunicación con otros

sistemas. En particular, es muy importante incorporar un acceso a las bases de datos de diferentes catálogos estelares para así poder localizar objetos astronómicos varios en forma rápida y consistente. Por otra parte, *TheSkyX* mejora el apuntamiento hacia los astros vía TPOINT [14]. Para realizar todas estas mejoras fue necesario identificar los comandos de bajo nivel que se transmiten mediante el puerto serial desde el programa TMOT hacia el controlador del NSH60, con la finalidad de poder aplicarlos en los métodos y propiedades que ASCOM define en su *standard* (Figura N° 7).

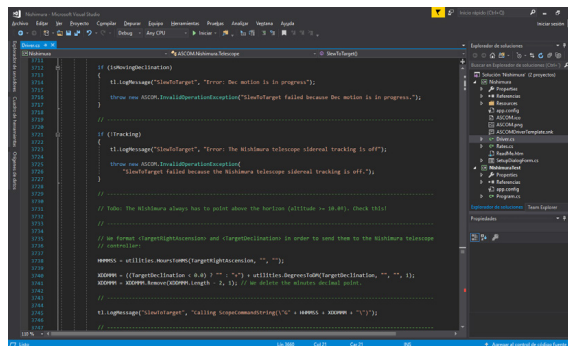


Figura 7: Extracto del método SlewToTarget de ASCOM.

2.7. Observación de eventos astronómicos durante el proceso de optimización del NSH60

Hemos utilizado el telescopio NSH60 en la observación científica de ciertos eventos astronómicos, entre ellos:

- La ocultación de la estrella UCAC4-406-125613 por el asteroide (15436) 1998 VU30, el 19 de junio de 2018. Ésta fue una de las primeras colaboraciones que se hicieron con la Dra. Myriam Pajuelo Cubillas, de la *Pontificia Universidad Católica del Perú*. La sombra del asteroide no terminó pasando por nuestra ubicación.
- El eclipse total de Luna ocurrido el 20-21 de enero de 2019. En esta oportunidad, se registraron datos espectrométricos de la región del cráter Tycho, a lo largo de las fases del eclipse, con la finalidad de determinar el espectro de transmisión de la atmósfera de nuestro planeta. Se está preparando un artículo científico sobre dicha observación.
- La ocultación de una estrella por una luna de Júpiter, el 3 de abril de 2019, apoyando un proyecto discreto del Dr. Erick Meza Quispe, quien actualmente desempeña labores en la *Comisión Nacional de Investigación y Desarrollo Aeroespacial*. Aquí, tuvimos inconvenientes por causa de la contaminación lumínica proveniente del horizonte este.
- Registro de datos astrométricos del asteroide 1999 KW4 (66391), los días 25 y 26 de mayo de 2019. Se contó con la colaboración del Observatorio y Planetario Moche, a cargo del Ing. Romel Villanueva Lujan, quien prestó sus periféricos astronómicos (equipos, entre cámaras y accesorios diversos) para ser acoplados al NSH60 en proceso de optimización. Es importante mencionar que se consiguió publicar un telegrama astronómico [15] conteniendo los resultados de las observaciones del asteroide.

3. CONCLUSIONES Y RECOMENDACIONES

El telescopio Nishimura NSH60 está en buenas condiciones operativas. Aun así, requiere la urgente implementación de un plan sistemático de limpieza y mantenimiento. Esto último para evitar un deterioro prematuro producido por el calor, la humedad y el polvo, entre otros factores. La posibilidad de re-ubicación

del instrumento es una opción posible ya que el NSH60 está siendo atacado por la humedad y el calor de la zona en la que actualmente se encuentra.

El *software* de enfoque del telescopio (FocaliS) desarrollado en este trabajo puede mejorar si incluimos el proceso de captura de imágenes dentro de un *thread* exclusivo que incremente la respuesta de nuestro sistema. Además, es posible robustecer el algoritmo de Levenberg-Marquardt usado en nuestro programa y agregar otro criterio de enfoque (como el *Half-Flux Diameter* [3]). Igualmente, el código desarrollado puede modificarse para mejorar su respuesta ante una estrella cuyo brillo pueda saturar el *imaging chip* de la cámara.

Se recomienda fortalecer las colaboraciones con otras instituciones que tengan un desempeño en astronomía, así como adquirir nuevos elementos de apoyo para la sala de control del NSH60 (una computadora y accesorios para el circuito de autoguiado, por ejemplo). Es importante la adquisición de mejores periféricos astronómicos (cámara CCD o CMOS de gama alta). Finalmente, se recomienda controlar la contaminación lumínica en el entorno de la *Estación Solar de Ica* dado que esto afecta directamente la calidad de las observaciones astronómicas realizadas con el NSH60.

4. AGRADECIMIENTOS

Agradezco al Instituto Geofísico del Perú y a mi asesor, por su paciencia, confianza y la oportunidad de poder brindar mis contribuciones a este proyecto. Asimismo, a las comunidades *ASCOM Initiative Discussion List*, *TeX-LaTeX Stack Exchange* y *Stack Overflow*, por su desinteresado servicio al responder muchísimas de las preguntas que tuve durante el desarrollo de mi trabajo. También, al Dr. José Ishitsuka Iba, por suministrarme una *laptop* apta para conectarse con la cámara CCD SBIG ST-7. Además, a la Mg. Victoria Gutiérrez Escate, por interceder a mi favor ante la empresa Nishimura y compartirme sus fotografías del momento mismo de la instalación del telescopio. Finalmente, a la astrónoma Shiomi Nemoto y al Ing. Taichi Tsujino, por ayudarme en puntuales traducciones del japonés al castellano.

5. BIBLIOGRAFÍA

- [1] Aperture Photometry Tool. URL: <http://www.aperturephotometry.org>
- [2] ASCOM. URL: <https://ascom-standards.org>
- [3] Diffraction Limited. Half-Flux Diameter (HFD). URL: <http://www.diffractionlimited.com/help/maximdl/Half-Flux.htm>
- [4] GeoGebra. URL: <https://www.geogebra.org>
- [5] Mutsumi Ishitsuka y col. Astronomy Against Terrorism: An Educational Astronomical Observatory Project in Peru. The Proceedings of the International Astronomical Union 8th Asian-Pacific Regional Meeting, Vol. I. Astronomical Society of the Pacific Conference Proceedings, Vol. 289. Editado por Satoru Ikeuchi, John Hearnshaw y Tomoyuki Hanawa. Mayo de 2003. 289...491
- [6] Jorge J. More. The Levenberg-Marquardt Algorithm: Implementation and Theory. Argonne National Laboratory, U. S. Energy Research and Development Administration. Conferencia de Análisis Numérico, Universidad de Dundee, Escocia, 28 de junio al 1 de julio de 1977. URL: <https://www.osti.gov/servlets/purl/7256021>
- [7] Nishimura. Hoja técnica Código N° KF60-001. Noviembre 1999
- [8] Nishimura. Hoja técnica Código N° KF60-003. Noviembre 1999
- [9] Gerald North. A better formula for telescopic limiting magnitudes? *Journal of the British Astronomical Association*, Vol. 107, No. 2, p. 82. Abr. 1997

[10] Charles Petzold. Programming Windows. The definitive guide to the Win32 API. Microsoft Programming Series. Microsoft Press. Quinta edición. 1998

[11] Santa Barbara Instrument Group. Application Note. Relay Adapter Box. Enero de 2007. URL: https://diffractionlimited.com/wp-content/uploads/2016/03/relay_box_instructions.pdf

[12] Santa Barbara Instrument Group. Operating Manual. CCD Camera Models ST-7E, ST-8E, ST-9E, ST-10E and ST-1001E. 2001. URL: <http://diffractionlimited.com/wp-content/uploads/2016/03/st78910man.pdf>

[13] Santa Barbara Instrument Group. SBIG Universal Driver/Library. Version 4.0. 17 de septiembre de 2001.

[14] TPOINT. URL: <http://www.tpointsw.uk>

[15] J. Ricra, M. Zegarra-Valles, N. O. Baella y R. Villanueva. Astrometric observations near the maximum approach of the Potentially Hazardous Asteroid (66391) 1999 KW4. Publicado en *The Astronomer's Telegram* (ATel # 13002), el 7 de agosto de 2019

PROCESAMIENTO Y PUESTA LÍNEA DE LOS DATOS DE RADAR PERFILADOR DE VIENTOS BLTR EN EL OBSERVATORIO DE HUANCAYO

**Processing and online visualization of the BLTR wind profiler at the Huancayo Observatory*

Jenny Margot Ramos Lazaro

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Físicas -Ingeniería Mecánica de Fluidos
jenny.ramos@jro.igp.gob.pe

Asesores:

Ing. Juan Carlos Espinoza
Dr. Danny Scipión

Radio Observatorio de Jicamarca - IGP

RESUMEN

En el presente proyecto se realizó la implementación de una aplicación web, que permite la visualización de los datos diarios obtenidos por el radar perfilador de vientos BLTR (radar de capa límite y troposfera) en sus cuatro componentes (viento zonal, viento meridional, viento vertical y relación señal a ruido-SNR) y sus respectivos promedios horarios. Además, se creó una clase que permite determinar y eliminar los valores anómalos (outliers) dentro de la muestra además de graficar los promedios horarios, usando la biblioteca de procesamiento de datos Signal Chain desarrollada en Python.

Palabras clave: Outliers, aplicación web, radar y procesamiento de datos.

ABSTRACT

The present project consists on the implementation of a web interface, that allows the visualization of the daily data obtained by the BLTR wind profiler radar (Boundary Layer and Tropospheric Radar) in its four components (zonal wind, meridional wind, vertical wind, and signal-to-noise ratio SNR) and their corresponding hourly averages. In addition, a class was created to identify and eliminate the outliers within the sample as well as to plot the hourly averages, by using the Signal Chain processing library developed in Python.

Keywords: Outliers, web application, radar and data processing.

1. INTRODUCCIÓN

El radar BLTR (radar de capa limite y troposfera) forma parte del Laboratorio de Microfísica Atmosférica y Radiación (LAMAR) ubicado en el Observatorio de Huancayo (HYO), laboratorio implementado con el objetivo de desarrollar investigación sobre Física Atmosférica en los Andes del Perú. Este radar esta configurado, para muestrear en más de un modo. Uno de ellos es el "modo bajo," donde el pulso de energía transmitido por el equipo puede ser de 60 m en longitud. La longitud del pulso determina la profundidad de la columna de aire que está siendo muestreada y la resolución vertical de los datos. En un "modo alto," la longitud del pulso se incrementa, usualmente a 100 m o mayor. Un pulso más grande significa que más energía está siendo transmitida en cada muestra, mejorando la *relación señal/ruido* (SNR, o S/R) de los datos. Usando una longitud de pulso mayor se incrementa la profundidad del volumen muestreado, y a su vez decrece la resolución vertical de los datos.

Radar BLTR registra datos de *velocidad* y *dirección* del viento a varias elevaciones sobre el terreno, estos datos son almacenados en una computadora pero la capacidad de almacenamiento del disco duro, no es suficiente para toda la data registrada. Por tal razón se plantea la implementación y creación de una aplicación web, que permita al usuario acceder a las imágenes procesadas de los registros captados por día en cada uno de los canales existentes (viento zonal, viento meridional, viento vertical y SNR), además haciendo uso del software de procesamiento de datos Signal Chain se implementara una clase que permita determinar y eliminar los outliers (valores anómalos) dentro de la muestra usando el método de las Desviaciones típicas. A su vez se realizará creación de gráficos, de promedios horarios en cada uno de los canales respectivos.

2. DESARROLLO

2.1. Metodos para la detección de outliers

2.1.1. Método del rango intercuartílico IQR (o rango intercuartil)

Es una estimación estadística de la dispersión de una distribución de datos. Consiste en la diferencia entre el **tercer** y el **primer cuartil**. Mediante esta medida se eliminan los valores extremadamente alejados. El rango intercuartílico es altamente recomendable cuando la medida de tendencia central utilizada es la *mediana* (ya que este estadístico es insensible a posibles irregularidades en los extremos).[1]

$$IQR = Q_3 - Q_1$$

$$valormin = (Q3 - 1.5 * (Q3 - Q1)) \dots \dots (I)$$

$$valormax = (Q3 + 1.5 * (Q3 - Q1)) \dots \dots (II)$$

Por lo tanto, en distribuciones con una gran *asimetría*, (alejadas de la distribución normal o campana de Gauss) es más apropiado medir la *tendencia central* y la *dispersión* mediante la *mediana* y el **rango intercuartil** respectivamente que con la *media aritmética* y la *desviación típica*. Con el **IQR** podremos elaborar los *diagramas de caja*, que es un instrumento muy visual para evaluar la dispersión de una distribución. Además, se suele crear un histograma para evaluar la forma y dispersión de los datos. Los histogramas funcionan mejor cuando el tamaño de la muestra es mayor que 20. [1]

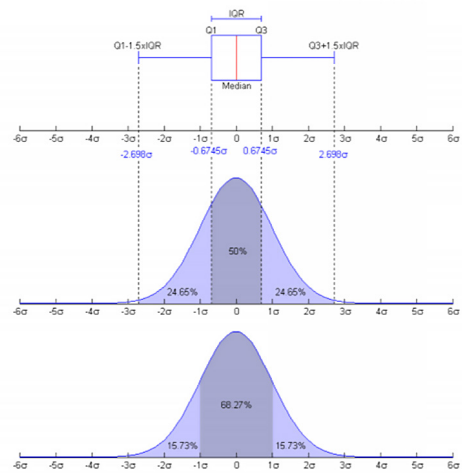


Figura 1. Histograma de los datos con una curva normal sobrepuesta para examinar la normalidad de los datos. Una distribución normal es simétrica y tiene forma de campana, como lo indica la curva. Fuente: Blog de Sandrine, consultora Senior Black Belt Six Sigma y Lean.

2.1.2 Método de las desviaciones típicas

Método basado en la desviación típica, considerara outliers a todos los datos fuera del intervalo [2]

Si consideramos $k=3$

valorminimo

$$= media - 3$$

• *Desviacion típica*(I)

valormaximo=

$$media + 3 \cdot \text{Desviacion típica} \dots \dots (II)$$

$$(media - 3 \cdot \text{Desviacion típica}, (media + 3 \cdot \text{Desviacion típica}))$$

$$s = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}$$

.....(III)

Desviación estándar

Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como Modelo–vista–controlador. Fue desarrollado en origen para gestionar varias páginas orientadas a noticias de la World Company de Lawrence, Kansas, y fue liberada al público bajo una licencia BSD en julio de 2005; el framework fue nombrado en alusión al guitarrista de jazz gitano Django Reinhardt. [3]

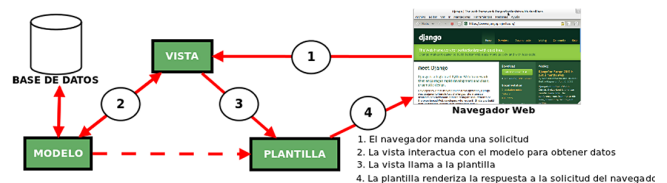


Figura 2. Estructura de Django. Fuente: Documentación de Django

a. Archivos del proyecto

- **__init__.py:** Este es un archivo vacío que le dice a Python que debe considerar este directorio como un paquete de Python. [3]
- **manage.py:** Este archivo contiene una porción de código que permite interactuar con el proyecto de Django de muchas formas. [3]
- **settings.py:** Este archivo contiene todas las configuraciones para el proyecto
- **urls.py:** Contiene las rutas que están disponibles en el proyecto, manejado por URLConf. [3]

b. Archivos de la aplicación [3]

- **__init__.py:** La misma descripción anterior (líneas arriba).
- **models.py:** En este archivo se declaran las clases del modelo.
- **views.py:** En este archivo se declaran las funciones de la vista.
- **test.py:** En este archivo se declaran las pruebas necesarias para la aplicación

2.6. Base de datos MongoDB

MongoDB es una base de datos orientada a documentos. Esto quiere decir que, en lugar de guardar los datos en registros, guarda los datos en documentos. Estos documentos son almacenados en BSON, que es una representación binaria de JSON.[4]

Una de las diferencias más importantes con respecto a las bases de datos relacionales, es que **no es necesario seguir un esquema**. Los documentos de una misma colección - concepto similar a una tabla de una base de datos relacional -, pueden tener esquemas diferentes. MongoDB está escrito en C++, aunque las consultas se hacen pasando objetos JSON como parámetro. Es algo bastante lógico, dado que los propios documentos se almacenan en BSON. [4]

3. RESULTADOS

3.1. Datos

Los archivos de datos de Análisis de medición de viento espaciado del sensor (SSWMA) están escritos en un formato binario estos fueron registros por día obtenidos desde 1 enero del 2016 hasta el 28 septiembre del 2017, en el modo de operación del radar bajo almacenados en los cuatro canales (viento zonal, viento meridional, viento vertical y SNR) de 132 alturas y 24 horas.

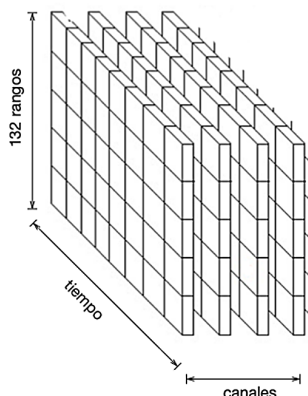


Figura 3. Estructura de Datos

3.2. Detección y eliminación de outliers

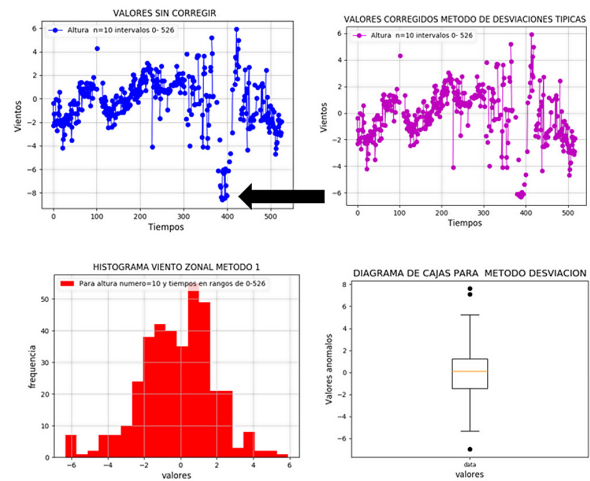


Figura 4. Observamos que el método De Desviaciones típicas, elimina tres valores ya que estos se encuentran fuera de su límite inferior -6.4, además corroboramos con el histograma que en dicha altura la muestra tiene una distribución normal, con el diagrama de cajas encontramos dos valores anómalos en la muestra.

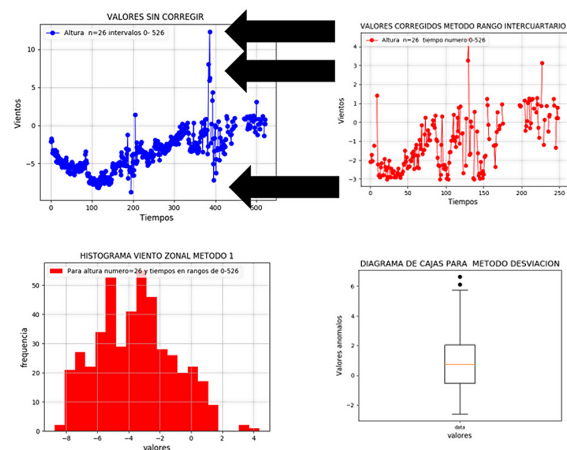


Figura 5. Observamos que el método de Rango Intercuartario, elimina 7 valores en la altura 26 en un rango de 526 tiempos, además corroboramos con el histograma que en dicha altura la muestra tiene una distribución normal, con el diagrama de cajas encontramos dos valores anómalos en la muestra lo que no concuerda con la eliminación dada.

3.2.1 Resultados detección y eliminación de outliers

EL método del rango intercuartario presentaba las siguientes inconsistencias:

- Al calcular Q1 y Q3, la presencia de valores no válidos (NaN), daba como resultados el valores que no podían considerarse validos para la determinación de el intervalo de mayor y menor velocidad.
- Los valores a eliminar por dicho método representaban porcentajes muy altos dentro del total de la muestra.
- En muchos casos los diagramas de cajas, no mostraban la misma cantidad de valores anómalos.

El método de desviaciones típicas, fue el elegido para realizar la eliminación de outliers debido a su mayor confiabilidad y al no involucrar en el calculo directamente a los valores NaN, aunque

para determinar la media y la desviación estándar de la muestra si se consideraban los valores anteriormente mencionadas.

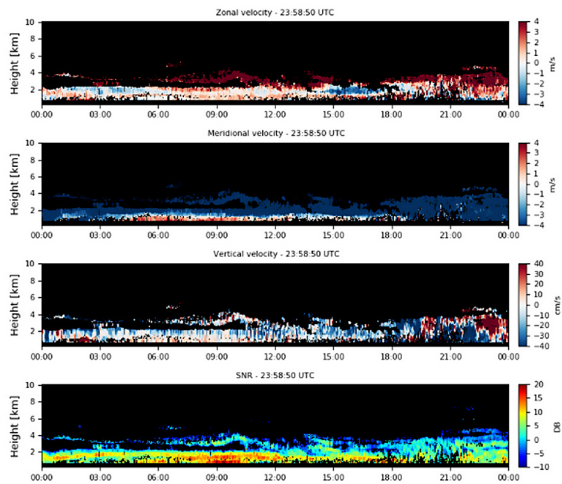


Figura 6. Datos sin corrección. Fecha 03.07.2020

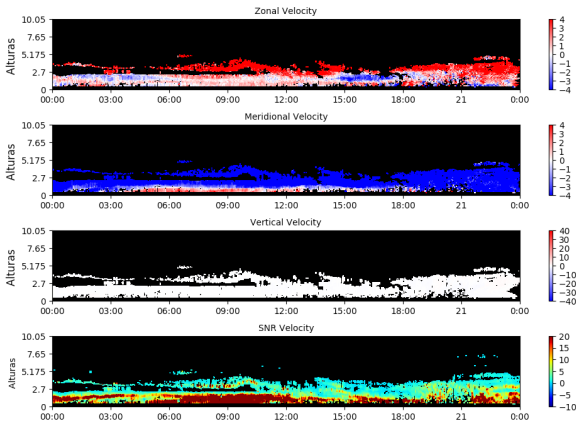


Figura 7. Datos corregidos. Fecha 03.07.2020

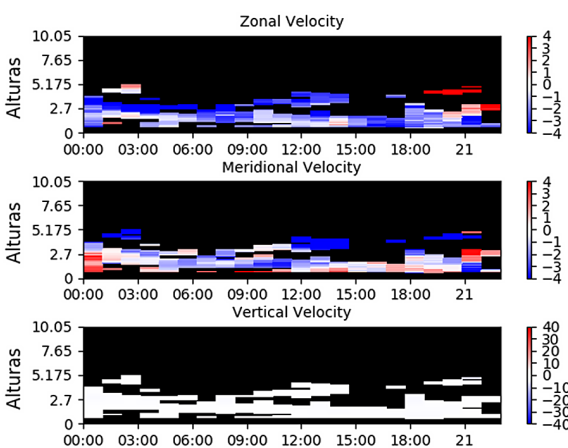


Figura 8. Promedios horarios. Fecha 03.07.2020

3.3. Aplicación Web

Se implementó una base de datos MONGODB llamada "dpbx" en donde se almacenaron los archivos contenido los datos corregidos y los promedios horarios.

El código de Django está escrito usando principios y patrones de diseño para fomentar la creación de código mantenible y

reutilizable.

Aplicación creada contiene la data corregida a través del método de Desviaciones típicas además de los promedios horarios. Se creó un calendario basado en JavaScript, que conecta con la base de datos dpbx creada en mongodb. El proyecto Django tiene la siguiente estructura:

- **__init__.py:** Este es un archivo vacío que le dice a Python que debe considerar este directorio como un paquete de Python.
- **manage.py:** Este archivo contiene una porción de código que permite interactuar con el proyecto de Django de muchas formas.
- **settings.py:** Este archivo contiene todas las configuraciones para el proyecto incluyendo la conexión de datos, además la ruta de los archivos estáticos de tipo javascript (donde se encuentra guardado en calendario generado), la misma carpeta contiene los archivos css que son los estilos conectados a la vista. Además, conecta con la carpeta Templates, que contiene las vistas, que son cuatro archivos de tipo HTML: home.html, realtime.html, databe.html y contact.html.
- **urls.py:** Contiene las rutas que están disponibles en el proyecto, manejado por URLConf.

c. Archivos de la aplicación realtime

- **models.py:** En este archivo se declaran las clases del modelo PXData. Que son dos datos de tipo Textfield de tipo Fecha que van a servir para realizar la búsqueda en la base de datos.
- **views.py:** En este archivo se declaran las funciones de la vista.

Tiene una conexión con los templates generados que se muestran en la aplicación.

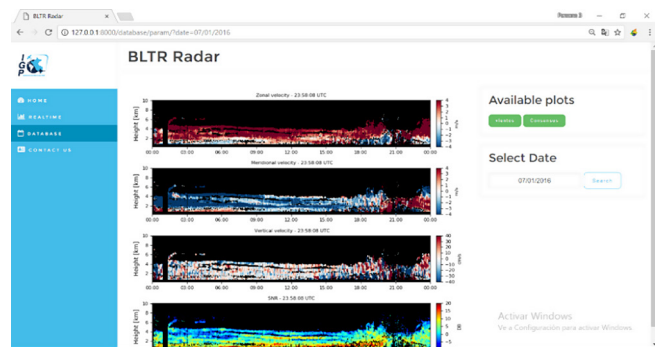


Figura 9. Consulta online para la categoría de datos del día 07/01/2016

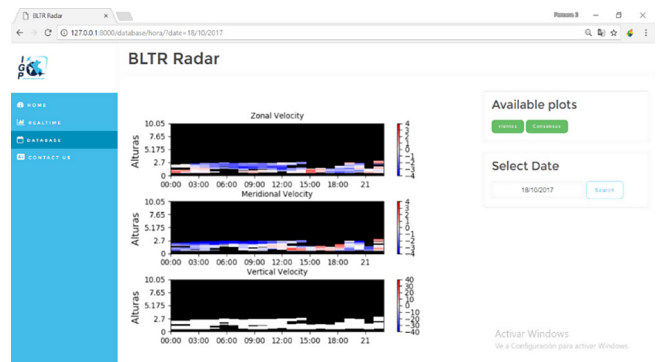


Figura 10. Consulta similar para el promedio diario

Beneficios de usar DJANGO

- El hecho de que esté escrito en Python permite que sea fácil de utilizar tanto para programadores expertos como para los que se están iniciando en el mundo de la programación.
- Maneja en su arquitectura el modelo MVC (Modelo-Vista-Controlador).
- Existe mucha documentación amigable y muy completa, tanto en inglés como en español.
- Cuenta con una API de base de datos robusta, también con un mapeador objeto-relacional.
- Tiene un despachador de URLs basado en expresiones regulares.

3. DISCUSIÓN

La presencia inicial de los valores anómalos, luego a ser eliminada por el método de desviaciones típicas, pero existen innumerables métodos para la eliminación de valores anómalos, como lo son el método de Chauvet y la distancia de Mahalanobis, la implementación de la misma se puede realizar en un futuro.

Si bien es cierto el framework Django facilita la creación de aplicaciones web, en la actualidad el uso de lenguajes como php y javascript, para realizar proyectos web, es mucho más común en la comunidad de desarrolladores.

4. CONCLUSIONES

El método del rango intercuartario eliminaba valores, al calcular Q1 y Q3. La presencia de valores NaN, daba como resultado valores que no podían considerarse válidos para la determinación del intervalo de mayor y menor velocidad.

Finalmente, se corrigieron los valores anómalos usando el método de las desviaciones típicas debido a que los valores eliminados eran menores. Además, que las muestras tomadas, cumplan con tener una distribución normal, esto se prueba al observar los histogramas ploteados.

Se logró realizar los gráficos de resumen horario para las velocidades Zonal, Meridional y Vertical.

Se logró implementar una base de datos en MongoDB con los gráficos generados.

El framework Django facilitó la conexión nos permite, obtener gráficos al día y el resumen horario de cada fecha.

Usar MongoDB, nos facilitó el manejo de documentos de una misma colección pueden tener esquemas diferentes.

5. RECOMENDACIONES

Se recomienda la recolección de más datos para ser almacenada en la base de datos creada.

Al reusar la aplicación, se tiene que tener en cuenta la versión instalada y los requerimientos, expresados en el archivo requirements.txt.

6. AGRADECIMIENTOS

Mi más sincero agradecimiento al Ing. Juan Carlos Espinoza y al Dr. Danny Scipión por su gran apoyo y comprensión, que me permitieron llevar a cabo el proyecto.

Mi profundo agradecimiento al Instituto Geofísico de Perú a través del Radio Observatorio de Jicamarca, por haberme dado la oportunidad de pertenecer a su programa de Prácticas Pre-Profesionales.

7. BIBLIOGRAFÍA

[1](Pérez 1987, Rousseeuw y Van Zomeren 1990) Comparación de métodos de detección de datos anómalos multivariantes mediante un estudio de simulación. . Universidad de Oriente, Núcleo de Nueva Esparta, Departamento de Estadística.

[2] Daniel Peña (2002) "Análisis De Datos Multivariantes". España s.a. mcgraw-hill / Interamericana de España.

[4]Django "Documentation". Django Project <https://www.djangoproject.com/>(visitada marzo 3 2018).

[5]MongoDB "mongodb 3.6 manual"<https://www.mongodb.com/es>(visitada marzo 3 2018)

RADIO ENLACE PARA LA AMPLIACIÓN DEL SISTEMA DE MONITOREO DE HUAICOS

**Radio link for the extension of the Huaicos Monitoring System*

Freddy Orlando Jara Poma

Universidad Nacional Mayor de San Marcos
Facultad de Ingeniería Electrónica y Eléctrica
freddy.jara@jro.igp.gob.pe

Asesor: Luis Tejeda

Radio Observatorio de Jicamarca - IGP

RESUMEN

Los huacos son producidos por fenómenos climatológicos, entre ellos el Niño Costero, y generan violentas inundaciones producto del desborde de los ríos y causan enormes desastres a su paso. Éstos ocurren con mayor intensidad en zonas cercanas a los ríos, en temporada de lluvias, como es el caso del Instituto Geofísico del Perú en su sede científica el Radio Observatorio de Jicamarca (ROJ). Los huacos vienen siendo uno de los fenómenos con mayor índice de destrucción en nuestro país. En temporada de lluvias, en el año 2017, se produjeron deslizamientos de lodo en la quebrada del río Huaycoloro en el distrito de Huachipa y en el distrito de Chosica, por las intensas lluvias, según informó el Servicio Nacional de Meteorología e Hidrología (Senamhi). Es por esa razón que se necesita ampliar los puntos de monitoreo mediante radio enlaces para una acción rápida en caso de desastres.

Palabras clave: Radio enlace, huacos.

ABSTRACT

Huacos are produced by climatological phenomena, like the Coastal El Niño, generate violent floods as a result of the overflowing of the rivers, and cause enormous disasters along their path. They occur with greater intensity in areas near the rivers, during the rainy season, as is the case of the the Geophysical Instituto of Perú at its scientific facility, the Jicamarca Radio Observatory (JRO). Huaicos are one of the phenomena with the highest rate of destruction in our country. In the rainy season of 2017, mud slides occurred in the Huaycoloro river ravine in the district of Huachipa and in the district of Chosica, due to heavy rains, according to the National Service of Meteorology and Hydrology (Senamhi). It is for this reason that an increase of monitoring points is needed with radio links for rapid action in case of disasters.

Keywords: Radio link, huacos.

1. INTRODUCCIÓN

El propósito de este proyecto es ampliar el alcance del SMH, que actualmente cuenta con dos puntos de monitoreo cercanos al IGP-ROJ, con un radio enlace que tenga un alcance promedio a 1 Km. Se realizaron pruebas iniciales para conocer el funcionamiento de los equipos y probar la capacidad de los mismos, dicha prueba fue a una distancia de 300 metros. El propósito de este radio enlace es establecer una conexión entre los dos puntos para un monitoreo constante con el suficiente ancho de banda para soportar los datos enviados de sensores y una cámara IP. Luego se realizaron pruebas finales entre dos puntos separados aproximadamente a 1 Km. En la Figura 1 se observa el esquema de conexión del enlace punto a punto en términos generales, el punto remoto tendrá los sensores.



Figura 1: Esquema general de conexión PtP.

2. DESARROLLO

2.1. Descripción de equipos

Para la implementación de un radio enlace se usaron dos equipos UBIQUITI NanoStation M2 los cuales funcionan a 2.4 GHz (banda no licenciada), Figura 2, estos equipos son especiales para zonas donde existe mucha corrosión eólica, como ROJ. Un equipo será configurado en modo PUNTO DE ACCESO y el segundo en modo ESTACIÓN. En la Tabla 1 se mencionan las principales características de los equipos, extraídas del Datasheet proporcionada por el fabricante. Vemos que el consumo máximo de solo la antena es de 8 W lo que produciría un consumo de energía máximo de 5.76 kWh al mes.



Figura 2: Equipo UBIQUITI NanoStation M2.

Los equipos UBIQUITI NanoStation M2 tienen dos puertos Ethernet y un botón de reset, como se muestra en la Figura 3 cuyos propósitos son: Puerto Ethernet Main, utilizado para conectar la alimentación y debe estar conectado a la LAN y al servidor DHCP. Puerto Ethernet Secondary, utilizado para puentear y admite paso PoE configurable por software. Botón Reset, para restablecer los valores predeterminados de fábrica, se debe mantener presionado el botón durante más de 10 segundos, mientras el equipo NanoStation ya esté encendido. Alternativamente, un botón de reinicio remoto se encuentra en la parte inferior del adaptador PoE.



Figura 3: Puertos y botón de reinicio (Reset) del equipo UBIQUITI NanoStation M2.

Al momento de realizar las pruebas de campo se necesitarán dos laptops para poder medir la capacidad del enlace y simular que existen sensores enviando información al servidor haciendo uso del programa Jperf del cual se hablará más adelante.

2.2. Configuración de equipos

Como parte de la instalación del Hardware de los equipos UBIQUITI NanoStation M2, se deben conectar el Puerto Main del equipo con el Puerto PoE del adaptador mediante un cable Ethernet, luego hacer una conexión del puerto LAN de una PC a la interfaz LAN del Adaptador PoE, luego el adaptador conectarlo a la alimentación 220 VAC tal como muestra la Figura 4.

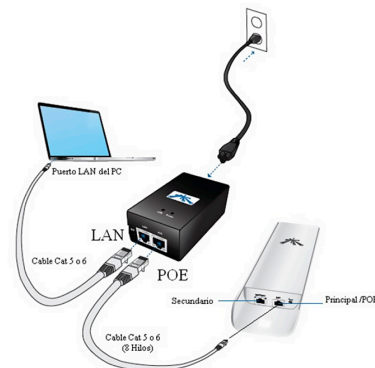


Figura 4: Instalación correcta del equipo UBIQUITI.

Luego de haber realizado una correcta instalación del hardware en los dos equipos se procede a configurar el software para poder realizar una conexión punto a punto. Para iniciar la configuración necesitamos conectarse a la interfaz de configuración de airOS. Debemos configurar el adaptador Ethernet en el sistema con una dirección IP estática en la subred 192.168.1.x, para ello debemos iniciar el navegador web y escribir <https://192.168.1.20> en el campo de dirección.

Para establecer una conexión punto a punto debemos configurar un equipo en modo Access Point y el otro en Station, a continuación se mostrará cómo realizar dicha configuración en ambos equipos.

2.2.1. Configurando equipo en modo Access Point

Configuración del equipo que estará en modo Access Point cuya MAC WLAN es 68:72:51:72:B0:76 el cual será configurado con una IP: 192.168.1.3 y será el encargado de recibir los datos

enviados desde el punto remoto, el cual tendrá los sensores funcionando.

Para asignar una nueva IP al equipo seleccionamos: Network >> Gestión >> Estática. Dirección IP >> 192.168.1.3.

Configuramos la parte inalámbrica del equipo: Wireless >> Modo inalámbrico >> Punto de acceso. SSID >> AAAAA (Nombre de la red). Ancho del canal >> 20 MHz. Lista de frecuencias >> 2412 MHz (puede usarse cualquiera de los canales). Potencia de salida >> 9 dBm.

2.2.2. Configurando equipo en modo Station

Configuración del equipo que estará en modo Station cuya MAC WLAN es 68:72:51:72:A6:E0 el cual será configurado con una IP: 192.168.1.2 y será el encargado de enviar los datos desde el punto remoto.

Para asignar una nueva IP al equipo seleccionamos: Network >> Gestión >> Estática. Dirección IP >> 192.168.1.2.

Configuramos la parte inalámbrica del equipo: Wireless >> Modo inalámbrico >> Station. SSID >> Seleccionar >> Buscamos el nombre de red elegida en el anterior equipo. Ancho del canal >> 20 MHz. Lista de frecuencias >> 2412 MHz. Potencia de salida >> 9 dBm.

Luego de haber configurado los dos equipos pasamos a dibujar, en la Figura 5, nuevamente el esquema de la conexión P2P con sus respectivas direcciones IP.

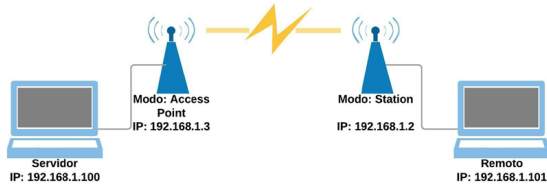


Figura 5: Esquema conexión P2P.

2.3. Uso de Jperf

JPerf es una interfaz gráfica para la popular herramienta de prueba de red Iperf. Con JPerf puede probar rápidamente una conexión WAN o LAN para determinar el rendimiento máximo de la red. Los resultados de las pruebas se grafican automáticamente y se presentan en un formato que es fácil de leer. JPerf está diseñado para ejecutarse como una aplicación cliente/servidor. Para ejecutar una prueba, se debe configurar un servidor JPerf en su red. Luego puede ejecutar un cliente JPerf desde otra ubicación en la red que se conectará al servidor remoto. En este proyecto se usará el protocolo UDP para el envío de datos para así poder medir la capacidad máxima del enlace.

2.4. Consumo aproximado de ancho de banda

Debido a que el enlace debe soportar los datos enviados por una cámara IP y algunos sensores instalados para monitorear la zona, se estimó un ancho de banda que pueda estar disponible para estos equipos.

Para la cámara IP podemos usar el códec H.264 con una resolución de 1280x720 que tendría una velocidad de bits máxima de la carga útil de video de 2500 Kbps y una velocidad de bits mínima de la carga útil de video de 500 Kbps. [4]

2 sensores de Ultrasonidos (SRF485WPR) de la marca Devantech tienen una interfaz de salida 485, se calcula que el tráfico aproximado podría ser de 128 Kbps (2x64 Kbps). [5]

Como capacidad total del enlace tenemos 2628 Kbps, si aseguramos este ancho como el 50% de la capacidad total del

enlace para mayor fiabilidad, tendríamos un enlace con un ancho de banda de 5256 Kbps.

2.5. Cálculo para la alimentación del sistema en un punto remoto

Habiendo realizado estos cálculos podemos calcular la capacidad necesaria que debe tener nuestra batería. El rendimiento de las baterías suele ser del 50%, si la energía que se consumirá es de 294 Wh, suponiendo el rendimiento antes mencionado habría un nuevo consumo de 588 Wh, de igual manera los inversores son necesarios para poder suministrar los 220 VAC que requiere el equipo UBIQUITI, estos inversores suelen tener un rendimiento del 10%, si añadimos este nuevo rendimiento tendremos un nuevo consumo de 646.8 Wh. Si adquirimos una batería de 12 VCC podemos obtener la capacidad en Ah. El consumo de 646.8 Wh dividido entre los 12 voltios de salida obtenemos la capacidad de 53.9 Ah.

3. RESULTADOS

En el desarrollo del proyecto se realizaron dos pruebas, la primera a una distancia aproximada de 300 metros y la segunda a una distancia aproximada de 1 kilómetro. En esta sección se mostrará las pruebas realizadas con el medidor de ancho de banda Jperf y sus respectivas gráficas.

3.1. Enlace de prueba a 300 metros

Como primera prueba se realizó un radio enlace entre el puesto de vigilancia del ROJ y el centro de operaciones del Radar SOUSY, los cuales se distancian aproximadamente unos 300 metros y presentan una línea de vista libre para que la primera zona de Fresnel no tenga obstrucción. A esta distancia la zona de Fresnel tiene un máximo de 3 metros. En la Tabla 1 se muestran las coordenadas de los puntos de prueba.

Coordenadas GSM	Latitud	Longitud
Vigilancia	11°57'11.06"S	76°52'39.64"O
SOUSY	11°57'11.63"S	76°52'30.01"O

Tabla 1. Coordenadas del enlace a 300 metros.

3.1.3. Pruebas con Jperf

Se realizó una prueba en Jperf de 18000 segundos enviando datos usando una capacidad de 20 Mbits/sec. En la Figura 6 se muestran los datos recopilados cada 10 minutos, mostrando la capacidad del canal vs el tiempo.

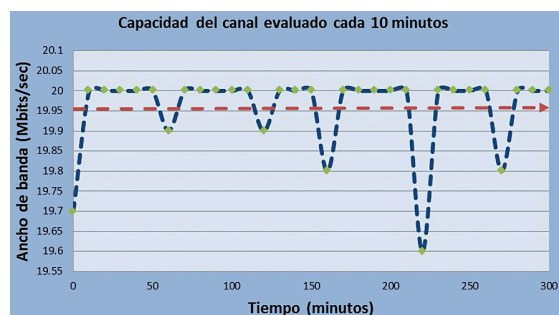


Figura 6: Capacidad del enlace a 300 metros.

3.2. Enlace de prueba a 1 Kilómetro

Como última prueba se realizó un radio enlace entre el mirador de la antena principal del ROJ y el punto de monitoreo de Río Seco, los cuales se distancian aproximadamente 1 kilómetro y presentan una línea de vista libre para que la primera zona de Fresnel no tenga obstrucción. A esta distancia la zona de Fresnel tiene un máximo de 5.6 metros. En la Tabla 2 se muestran las coordenadas de los puntos de prueba.

Coordenadas GSM	Latitud	Longitud
Mirador	11°57'7.22"S	76°52'33.20"O
Río seco	11°56'51.19"S	76°52'4.26"O

Tabla 2. Coordenadas del enlace a 1 Km.

3.2.4. Pruebas con Jperf

Se realizó una prueba en Jperf de 17040 segundos enviando datos usando una capacidad de 50 Mbps. En la Figura 7 se muestran los datos recopilados cada 4 minutos, mostrando la capacidad del canal vs el tiempo.

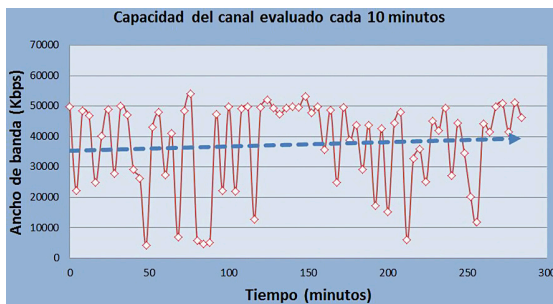


Figura 7: Capacidad del enlace a 1 Km.

4. DISCUSIÓN

Se obtuvieron dos secuencias de datos, una a 300 metros la cual se realizó pruebas a 20 Mbps y dieron resultados muy estables sin muchas variaciones debido a la distancia que separaba a los equipos y que tenían una línea de vista libre de obstáculos, luego en la segunda prueba a 1 kilómetro se aumentó la capacidad de transmisión de datos a 50 Mbps para tener un ancho de banda suficiente para soportar cámaras IP y sensores pero en esta prueba se tuvieron algunas caídas que en su mayoría no fueron menores a 20 Mbps dando como ancho de banda promedio aproximadamente 35 Mbps, en el trabajo de campo estas pruebas dependen mucho de ensayo y error debido a que tenemos que verificar el alineamiento correcto de las antenas de los equipos, contar con una línea de vista libre de obstrucciones así como un soporte del equipo lo suficientemente alto para evitar atenuaciones, los equipos de alimentación como la batería e inversor deben estar en correcto funcionamiento y lo suficientemente cargados para soportar el consumo de una laptop y del equipo UBIQUITI.

5. CONCLUSIONES

Para la realización del proyecto se tuvieron que analizar el funcionamiento de los equipos, la correcta configuración de los mismos. Para algunas aplicaciones, los fabricantes de los equipos brindan instrucciones sobre cómo configurar correctamente un equipo para una aplicación específica, de igual manera podemos ir cambiando la configuración para poder mejorarla de acuerdo

a nuestras necesidades del enlace.

El uso del programa Jperf resultó ser muy importante para medir la capacidad del enlace, este programa tiene distintas maneras de configurarse, para medir la capacidad del canal al máximo se usó el protocolo UDP pero se puede usar el protocolo TCP para otras aplicaciones que requieran el uso de este protocolo.

Al realizar el trabajo de campo se tiene que tener en cuenta la energía que consumen los equipos para poder elegir una batería y a la vez un inversor debido a que los equipos funcionan con alimentación alterna, así mismo ver los detalles del panel solar como fuente de alimentación para la batería.

6. RECOMENDACIONES

Para mejorar la calidad del enlace podría cambiarse la frecuencia de operación de los equipos para poder disminuir la zona de Fresnel y evitar algunas atenuaciones.

Poder adquirir un inversor nuevo para poder evitar posibles picos o caídas indeseadas alterando la transmisión o recepción de datos.

Es necesario leer acerca del funcionamiento del Jperf para poder tener una visión clara de que datos estamos recibiendo y que variaciones tienen.

Leer bien el manual de los equipos UBIQUITI para saber conocer cuál es la potencia máxima de transmisión y cuál es el consumo del equipo.

7. AGRADECIMIENTOS

Quiero empezar agradeciendo al equipo que hizo posible desarrollar este proyecto, ayudándome con la implementación de los equipos, con el transporte a los lugares donde se realizaron las pruebas, gracias por su paciencia y apoyo. De igual manera agradecer al Instituto Geofísico del Perú a través del Radio Observatorio de Jicamarca por haberme brindado la facilidad de usar los equipos necesarios para este proyecto el cual me ayudó mucho en mi formación profesional. Así mismo, brindar mi agradecimiento a mi asesor Luis Tejeda por su apoyo y orientación a lo largo del proyecto.

8. BIBLIOGRAFÍA

- [1] Ubiquiti Networks, "Quick Start Guide."
- [2] M. GARBARUK and N. LITWINCZUK, "DESIGN AND ANALYSIS OF RADIO LINKS FOR RADIOCOMMUNICATION SYSTEMS," 1973.
- [3] R. Díaz Vargas, "Diseño de radioenlace microondas isla san lorenzo – campus pucp para el proyecto peru magneto," pp. 52–53, 2015.
- [4] Microsoft, "Network bandwidth requirements for media traffic in Lync Server 2013," 24 Septiembre 2015. [En línea]. Available: [https://technet.microsoft.com/en-us/library/jj688118\(v=ocs.15\).aspx](https://technet.microsoft.com/en-us/library/jj688118(v=ocs.15).aspx).
- [5] A. N. Ince, "Principles of Integrated Maritime Surveillance Systems", Springer, 1999.

SISTEMA DE DETECCIÓN Y CONTROL DE FASE REMOTO PARA UN ARREGLO DE ANTENAS DIPOLO

**Remote phase detection and control system for a dipole antenna array*

Cesar Raul Alan Cruz Gutierrez

Universidad Nacional de Buenos Aires
Facultad de Ingeniería
cesaraulcg@gmail.com

Asesores: Juan Carlos Espinoza, Joaquín Verástegui

Área de Investigación Desarrollo e Innovación
Radio Observatorio de Jicamarca - IGP

RESUMEN

El radar principal del Radio Observatorio de Jicamarca cuenta con varios módulos que hacen posible la generación, emisión, recepción y almacenamiento de las señales. Una de las principales limitaciones de la antena es que, al ser una antena fija su cambio de ángulo de apunte solo puede ser realizado por cambios de fase manuales, para ello el área de I+D+I, ha venido desarrollando un módulo encargado de realizar cambios de fase variando las longitudes de cables, se tienen 4 bloques que contienen los relés internamente, separados por los cables de diferentes longitudes para realizar el cambio de fase en función a ello. Este cambio de fase debe ser controlado por un sistema embebido que a su vez debe estar conectado a un switch ethernet junto a los demás módulos "Antenna Beam Switching" (ABS). En ese contexto se plantea desarrollar un módulo de control y detección de fase, implementado en un sistema embebido capaz de mantener comunicación multicast por ethernet para mantener comunicación con el módulo principal de control, que se pueda replicar en todos los módulos de la antena y pueda brindar la información en tiempo real en el computador del módulo principal.

Palabras clave: ABS, RTOS, ARM, Sistema embebido, Fase, antena

ABSTRACT

The main radar of the Jicamarca Radio Observatory has several modules that make possible the generation, emission, reception, and storage of the signals. One of the main limitations of the phase-array antenna is the manual phase shifting capability. For this, I+D+I laboratory has been developing a module for phase shifting by switching among the different cable lengths. There are 4 blocks that contain the relays internally, separated by the different cable lengths to change the phase accordingly. This phase switching must be controlled by an embedded system which in turn must be connected to an ethernet switch next to the other ABS modules. In this context, it is proposed to develop a phase detection and control module, implemented in an embedded system capable of maintaining multicast communication via ethernet to maintain communication with the main control module, which can be replicated in all antenna modules and can provide real-time information on the main module computer.

Keywords: ABS, RTOS, ARM, Embedded system, Phase, antenna.

1. INTRODUCCIÓN

ABS se ha convertido en una herramienta importante para la antenna principal del observatorio, ya que permite generar mas experimentos y mejores estudios haciendo uso de la antenna, en tal sentido el area I+D+I del observatorio busca mejorar constantemente el sistema ABS, cuya variable principal es la fase, ademas de realizar los cambios de fase remotos bajo requerimientos del usuario el sistema ABS tiene varios componentes que ayudan a monitorear el correcto funcionamiento del sistema y monitorear la variable principal.

Además de los sensores y actuadores que son parte del sistema ABS, uno de los bloques principales del sistema es el control, en cuyo bloque estarán la comunicación, almacenamiento de apuntes, digitalización de sensores, algoritmo de control, entre otros que han venido siendo agregados para alcanzar las mejores respuestas de la variable principal ademas de el correcto funcionamiento, todas las funcionalidades y actualizaciones llevan a tener un sistema con buenas características de uso de recursos y manejo de tareas y en el presente proyecto la solución planteada para lograr la eficiencia y resiliencia del sistema, será implementada sobre un sistema Real Time Operative System(RTOS) en el sistema embebido de control, para ello es importante identificar los sub bloques y componentes que serán parte del sistema.

En la figura 1 vemos todos los bloques del sistema ABS como está implementado actualmente, donde estan los modulos de radio frecuencia y control, apartir de este podemos tener un diagrama general en figura 2, apartir del cual podemos extraer las variables y componentes necesarios para plantear un esquema para RTOS.

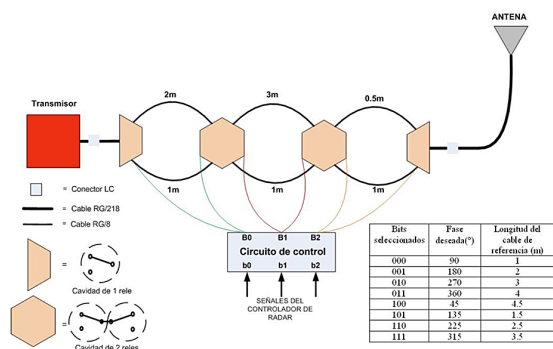


Figura 1. Diagrama de bloques RF+Control

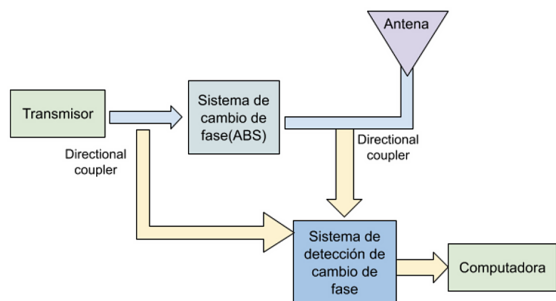


Figura 2. Diagrama de bloques general a partir del diagrama 1

2. TEORÍA INVOLUCRADA

2.1. Sistema operativo

A nivel de desarrollo en sistemas embebidos podemos decir.

Es un conjunto de programas que ayudan a gestionar los recursos

de hardware disponibles, entre ellos el tiempo del procesador y la memoria.

La gestión del tiempo del procesador permite al programador de aplicaciones escribir múltiples subprogramas como si cada uno fuera el único que utiliza la CPU.

Una parte del OS se encarga de asignar tiempo de ejecución a todos los programas que tiene cargados en base a un juego de reglas conocido de antemano. A estos subprogramas se los llama tareas

2.2. Porque el uso de un RTOS

- Un RTOS es un sistema operativo de tiempo real. Se utiliza cuando tiempo de respuesta ante ciertos eventos es un parámetro crítico.
- Cuando respuestas demasiado tempranas o muy tardías podrían ser indeseables ya que afectan las funciones del sistema.
- Un RTOS se emplea cuando hay que administrar varias tareas simultáneas con plazos de tiempo estrictos.
- Multitarea, escalabilidad, mayor reusabilidad.

Los puntos expuestos describen características que se espera que nuestro sistema cumpla y supere para su funcionamiento correcto

3. DESARROLLO

3.1. Bloques y capas de RTOS para nuestro sistema

Luego de justificar y resaltar la importancia de su implementación, se procedió a hacer un análisis de la información recabada en los documentos que describen el sistema actual de ABS y los requerimientos de sistema con el estándar IEEE 830.

Posterior al análisis se identifica los componentes de cada bloque Figura 3 posteriormente se plantea el esquema en capas Figura 4.

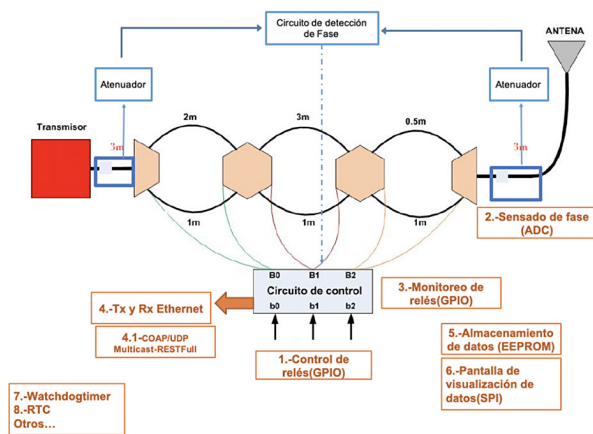


Figura 3. análisis de bloques

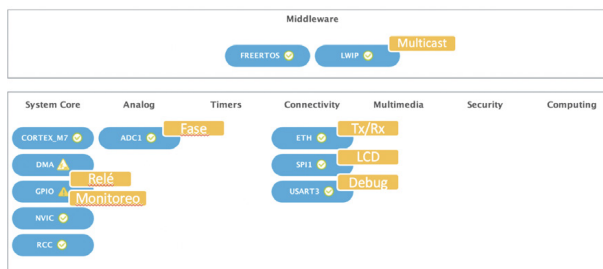


Figura 4. diagrama por capas

2.2. Prioridad y sincronía de tareas

En el flujo de tareas del sistema existen tareas críticas donde se envían o transmiten información importante o procesos críticos donde se procesan y/o calculan variables críticas, la naturaleza de un sistema RTOS hace que las diferentes tareas que tiene el sistemas puedan compartir el tiempo y recursos del procesador, bajo el propio criterio del sistema operativo es decir, el sistema operativo que se esta implementando elige que sistema se ejecuta, a nivel de desarrollador tenemos la capacidad de establecer prioridades de ejecución para ello se plantea las prioridades en la figura 6, esto en base a las necesidades encontradas y al criterio propio, bajo estos parámetros fijados el sistema operativo irá ejecutando las tareas.

Muchas de las tareas y bloques del sistema comparten información entre si o dependen el uno de otro, por ejemplo, en la figura 5 tenemos la tarea de censado de fase que en su primera etapa usa el bloque ADC del embebido, posteriormente hace un calculo del ángulo, ambas tareas deben estar en sincronía ya que el calculo depende del bloque ADC.

Hasta el momento son 8 tareas que se tienen sincronizadas, cambio de Fase transmisión y recepción, parte del desarrollo e implementación comprende el análisis y estudio de las tareas que y/o bloques que deben estar enlazados.

Sensado Fase	ADC		
	CALCULO		
Cambio Fase	CARGA MEM		
	SET GPIO		
Rx	ETHERNET	Multicast	Mutex_Rx
	LWIP		
Tx	ETHERNET	Multicast	Mutex_Rx
	LWIP		

Figura 5. Sincronización de tareas

Send_LCD	P=x+1
ADC	P=x+2
CALCULO	P=x+1
CARGA_MEM	P=x+2
SET_GPIO	P=x+3
ETHERNET	P=x+2

Figura 6. Prioridades de tareas

2.3. Tareas críticas con Mutex

En el punto 2.1.1 se mencionó la prioridad de tareas y se asignó una prioridad a cada tarea que hasta el momento se tiene estudiada, sin embargo existen tareas críticas donde se requiere que el sistema operativo no interrumpa el proceso ya que pueden existir perdida de datos o perdidas de sincronía, RTOS hace uso de mutex para estos escenarios, lo que nos permite mutex es indicar al sistema operativo donde inicia nuestra tarea crítica y donde termina, a nivel de código, para que en ese fragmento

el sistema operativo no disponga del CPU o recursos del sistema hasta que se culmine en la figura 5, podemos apreciar que se colocó mutex tanto en la transmisión como en la recepción, esto es en base al criterio de desarrollo y a la importancia que tiene este proceso dentro todo el sistema, en adelante se hará el uso de este recurso de igual manera, considerando si la tarea es crítica para el funcionamiento del sistema o no.

3. CONCLUSIONES

Este es un proyecto que aun no se culmina, las etapas faltantes tomaran de base los criterios hasta ahora tomados.

La implementación de un RTOS para el sistema de control ABS brindará mayor flexibilidad para futuras actualizaciones.

Las capacidades del microcontrolador ARM Cortex M7, nos permite cumplir los requerimientos actuales y pensar en actualizaciones futuras.

Será necesario el uso de software que facilite el estudio de los tiempos de tarea para seguir implementando los componentes del sistema.

4. RECOMENDACIONES

Mantener el modo de desarrollo basado en RTOS, será de vital importancia para cumplir con la escalabilidad y mantenimientos planteados al inicio, para ello se esta generando la documentación necesaria.

5. AGRADECIMIENTOS

Al Instituto Geofísico del Perú por brindarme las facilidades para desarrollar proyecto.

Al Dr. Marco Milla y al Ing. Juan Carlos Espinoza, por estar a cargo de mi capacitación en el Observatorio

6. BIBLIOGRAFÍA

Joseph Yiu, Definitive guide to ARM Cortex M3-M4.
 Informe IGP- ABS
 Introducción a los Sistemas operativos de Tiempo Real, Alejandro Celery - 2014
 Interrupciones - Ejemplos con FreeRTOS, CESE, Franco Bucafusco, 2017

SISTEMA DE SINCRONIZACIÓN DE PULSOS PARA VIPIR BIESTÁTICO

**Pulse Synchronization System for Bistatic VIPIR*

Edmundo Arom Miranda Zárate

Pontificia Universidad Católica del Perú
Facultad de Ciencias e Ingeniería
emirandaz@pucep.pe

Asesores: César de la

Jara, Joaquín Verástegui

CIELO - Radio Observatorio de Jicamarca
- IGP

RESUMEN

En este trabajo se explica el diseño y desarrollo de software y hardware de un Sistema de Sincronización de Pulsos para VIPIR Biestático capaz de generar pulsos de salida a cierta hora y con características de ancho y período que permitirán iniciar mediciones ionosféricas. El sistema está compuesto de un MCU y un FPGA. El MCU permitirá la interacción vía Ethernet, mientras que el FPGA será la generadora de pulso, utilizará un reloj externo de 10 MHz de un Trimble-GPS y la hora UTC que irá comparando con la hora programada. Además se ha diseñado un PCB como shield entre MCU-FPGA y que contenga el circuito acondicionador de reloj, las entradas y salidas.

Palabras clave: generadora de pulso, hora UTC, circuito acondicionador

ABSTRACT

In this paper, the design and development of software and hardware for a Pulse Synchronization System for Bistatic VIPIR capable of generating pulses at a certain time and with some other parameters as width and period to initialize ionospheric measurements is presented. The system is composed by an MCU and a FPGA. The MCU will enable the interaction via Ethernet, while the FPGA will be the pulse generator, it will use an external 10 MHz clock from de Trimble-GPS and the UTC time to compare with the set time. Besides, it has been designed a PCB as a shield between the MCU-FPGA and has the clock conditioning circuit, the input and outputs.

Keywords: Pulse generator, UTC time, conditioning circuit

1. INTRODUCCIÓN

El Instituto Geofísico del Perú, a través del Radio Observatorio de Jicamarca (ROJ), específicamente el área de CIELO (Cluster of Instruments for Equatorial and Low Latitude Observations) se encarga de operar diferentes tipos de radares para observaciones de la ionósfera a través de equipos geofísicos distribuidos en toda Latinoamérica. Para obtener mediciones más acertadas, es necesario que estén sincronizados con tiempos precisos. Habitualmente, cada equipo emplea dispositivos GPS que reciben la hora UTC satelitalmente. Puesto que la zona de la ionosfera está ubicada por encima de los 80 km y los equipos de por sí se encuentran bastante distanciados entre sí, las mediciones serán afectadas por la precisión del tiempo empleado para su sincronización.

El propósito de este proyecto es mejorar la conectividad entre ionosondas ubicadas en lugares distintos, desarrollando un sistema de sincronismo que permita transmitir y recibir ondas electromagnéticas desde distintos puntos (sistema biestático). De esta manera, se aprovechará mejor el potencial de LISN.

El sistema de sincronismo involucró tanto software como hardware y permitirá al operario de la sonda configurar la hora exacta para el inicio de mediciones vía Ethernet, de igual forma podrá ser replicado y fabricado para cada locación a enlazar.

El sistema cuenta con los siguientes requerimientos:

- El operario desde cualquier PC perteneciente a la misma red que la asignada al MCU podrá setear las configuraciones del PPS y del Pulso de salida vía Ethernet con comandos API REST
- El TinyFPGA será la unidad de procesamiento tanto del módulo divisor PPS como del generador de pulso para ello usará como señal de reloj para el FPGA los 10MHz del Trimble previamente acondicionados a 3.3V y convertidos a una señal cuadrada.
- Se propone una comunicación SPI entre MCU-FPGA compuesta de 4 canales (sin incluir GND) con un clock, mosi, miso y ss aprovechando los pines dedicados para SPI en el MCU y funciones de SPI, en el caso del FPGA utilizando un módulo SPI_slave descrito en Verilog.
- La comunicación entre FPGA y Trimble será UART a 9600bps con 1 bit de inicio, 1 bit de parada sin paridad.
- Se creará un backchannel UART de respaldo entre MCU-FPGA.

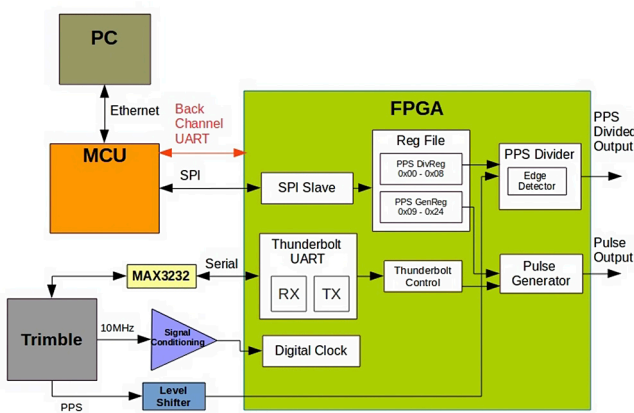


Figura 1. Diagrama de bloques del Sistema de Sincronización

2. DESARROLLO

2.1. Selección de Dispositivos

Los dispositivos principales empleados en este sistema de sincronización se detallan a continuación:

- Tiva TM4C129E Crypto

Tipo: MCU
 Marca: Texas Instruments
 Forma de cargar programas: MicroUSB
 Fuente de Alimentación: 3.3 V
 Frecuencia de Reloj Interno: 120 MHz
 IDE recomendado: Energia

- TinyFPGA BX

Tipo: FPGA
 Marca: TinyFPGA
 Forma de cargar programas: MicroUSB
 Fuente de Alimentación: 1.2 V
 Frecuencia de Reloj Interno: 16 MHz
 IDE recomendado: ICEcube2

- ThunderBolt E

Marca: Trimble
 Fuente de Alimentación: 24V
 Salidas: 10 MHz, PPS
 Comunicación: RS232

Revisando las especificaciones eléctricas del MCU y FPGA, se decidió tener una sola fuente de alimentación para el Tiva y que esta última suministre energía al TinyFPGA.

Con respecto al ThunderBolt ambas salidas son de 5V y los GPIO (General Purpose Input Output Pins) son de 3.3V es necesario adaptar ambas señales. Para el caso del PPS (Pulse per Second) se usó el buffer SN74LVC1G34 como level shifter de 5V a 3.3V. Por otro lado, dado que no es del todo recomendable conectar las salidas del TinyFPGA a otro lado fuera del PCB (en este caso se conectarán a uno de los módulos de VIPIR) se empleará el 74ACT541 como buffer intermediario.

Para la salida de 10 MHz ya que se trata de una señal sinusoidal y usar esta como señal de reloj para el FPGA. En la Figura 2 se propuso el siguiente circuito acondicionador de señal que se encarga de transformarla en una señal cuadrada de 0-3.3V compuesta de comparadores en histéresis.

Otro dato a considerar es que el módulo trimble envía y recibe información bajo la norma RS-232, de acuerdo a la Figura 3, el MAX3232 permitirá que la comunicación sea serial (UART a 9600bps). Este componente es alimentado con 3.3VDC. Su entrada provendrá del conector DB9 Male (que será soldado al PCB) y las salidas de transmisión y recepción (TX y RX) se conectarán directamente al FPGA.

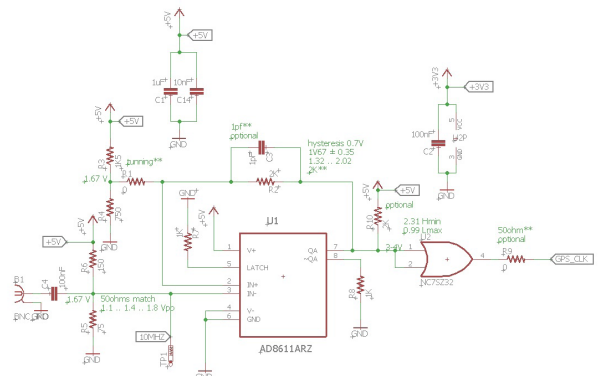


Figura 2. Esquemático del Circuito Acondicionador de Señal

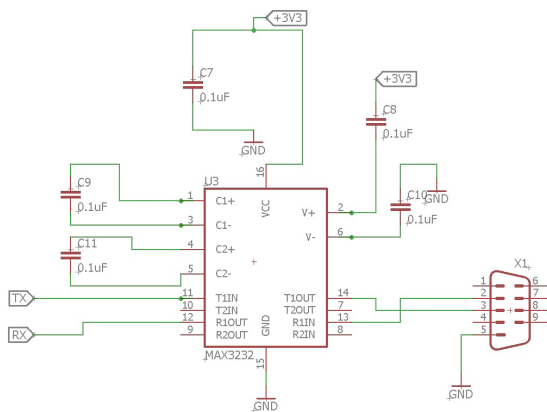


Figura 3 . Esquemático del MAX3232 y conector DB9 Male

2.2. Programación de los dispositivos

2.2.1. Tiva TM4C129E Crypto

El Tiva servirá como intermediario entre operario y FPGA, se aprovechará el puerto ethernet que facilitará las configuraciones remotas desde cualquier PC que pertenezca a la red del MCU. Para su programación, se usó ENERGIA que brinda las facilidades del Arduino Software para LaunchPads Tiva de Texas Instruments.

Las comunicaciones entre PC-MCU se harán mediante APIs-REST. Un **API-REST** (Application Program Interface - Representational State Transfer), en términos simples, es un conjunto de reglas y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas. En este caso se usarán comandos HTTP como GET (consultar y leer) y POST (para crear o escribir). Una ventaja de usar APIs-REST es que es independiente del tipo de plataformas o lenguajes, por tanto, son adaptables, lo único importante es mantener un formato estándar, en este caso en formato JSON. A continuación, se adjuntan los comandos API creados junto a la descripción de qué es lo que hacen.

Request	Type	Comment
reset	POST	Activates PIN for reset de FPGA
changeip	POST	Change the network configuration of the MCU Radar Controller
setdate	POST	Set date por Synchronization Pulse
ppsdivider	POST	Set configurations por PPS divider
getdate	GET	Get current time and date of GPS trimble

Tabla 1. Comandos APIs usados

Como se aprecia en la Tabla 1 el payload será enviado en JSON un formato de texto bastante sencillo de leer. En este caso se enviarán las configuraciones tanto como del pulso de sincronismo, del divisor de pps y de la configuración de IP bajo este formato.

```

SET DATE
{"year": [2019], "month": [3], "day": [7], "hour24h": [19], "minute": [40],
"second": [0], "widthPulse": [1000000], "period": [3000000],
"enable": [1]}
    
```

Figura 4. Ejemplo de configuración de comando setdate

2.2.2. TinyFPGA Bx

Se empleó el editor de texto ATOM y su complemento para archivos Verilog HDL para describir el hardware, como buena práctica se emplearon diferentes módulos con diferentes funciones particulares, como por ejemplo módulo SPI (para comunicación MCU-FPGA), módulo UART (para comunicación Trimble-FPGA), así como módulos exclusivos para describir la lógica de funcionamiento del divisor de PPS y el generador de pulso.

Para sintetizar los circuitos diseñados y módulos descritos en lenguaje Verilog se optó por usar el IDE oficial del producto llamado Lattice iCEcube2 disponible para Linux, este permite referenciar varios archivos Verilog y asignar prioridad al top o módulo principal del circuito. Este archivo "top" es el que llamará a los demás módulos programados (proceso similar a incluir librerías en un programa). Para ello es necesario incluir la ruta de los archivos secundarios de Verilog a usar. Para luego compilar todo el proyecto con ayuda de la herramienta Synplify Pro.

Como último paso, para cargar el programa al TinyFPGA, conectar vía MicroUSB y usar el programador para Linux TINYPROG. Para cada vez que se desee subir un programa el bootloader de arranque pedirá que se **reseteo físicamente** el FPGA. Luego de cargado el programa se podrá desconectar el cable MicroUSB y seguir con la alimentación externa de 5VDC suministrada, en este caso, por el MCU.

Como se recuerda la comunicación entre MCU-FPGA será SPI y cada parámetro de configuración tanto como del Pulso de Salida como del PPS dividido, serán almacenados en el FPGA en registros, por lo cual resulta útil definir una dirección a cada parámetro. Por tanto, el primer byte será la dirección del registro y el segundo byte el valor a guardar.

PPS divider		Pulse generator	
address	data	address	data
0x00	periodicTrue	0x09	year
0x01	divNumber	0x0A	year_MSB
0x02	phase_MSB	0x0B	month
0x03	phase	0x0C	day
0x04	phase	0x0D	hour
0x05	phase_LSB	0x0E	minute
0x06	width	0x0F	second
0x07	start	0x10	width_MSB
0x08	stop	0x11	width
		0x12	width
		0x13	width_LSB
		0x14	pulse_Enable
		0x15	periode_MSB
		0x16	periode
		0x17	periode
		0x18	periode_LSB

Tabla 2. Dirección y contenido de cada parámetro fijado

2.3. Diseño del PCB

Algunas consideraciones

- Los componentes serán de tipo SMD.
- Con un estudio previo del crosstalk producida por canales contiguos, se optó por usar los pines del TinyFPGA de forma intercalada, escogiendo como clock al pin 16 (recomendado por Lattice).
- Se aisló las entradas y salidas de RF a un lado del PCB alejado de la entrada digital del DB9 male para evitar interferencias.
- Se utilizó conector BNC-female para todas las entradas y salidas RF, menos para la del pulso de salida que usó un SMA-female, pues este se conectará al módulo de VIPIR con un cable SMA.

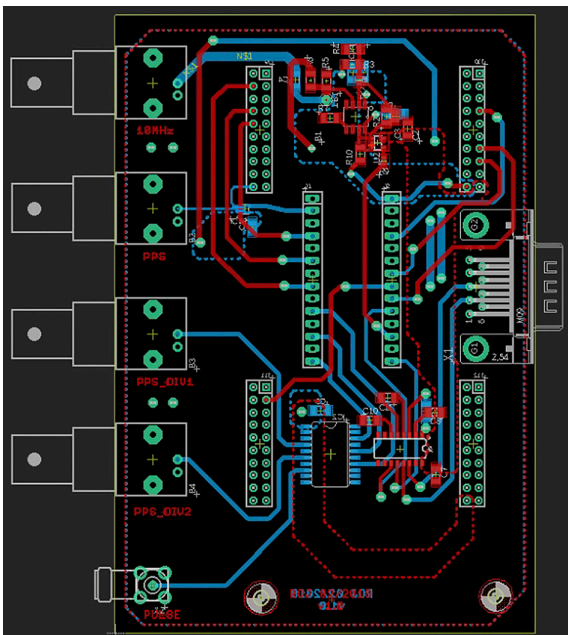


Figura 5. Diseño de la Placa componentes y pistas

3. RESULTADOS

El diseño y construcción del sistema de sincronización de pulsos se concluyó satisfactoriamente. Para su verificación se usaron test bench o simulaciones y comprobaciones mediante un analizador lógico.

3.1. Simulaciones o Testbench

Como primera prueba se realizaron archivos en verilog que probarán diferentes combinaciones para simular y verificar que el comportamiento del hardware descrito sea el correcto. La herramienta Modelsim permite compilar y simular archivos, así como representarlos en una onda de tiempo

La lógica del generador de pulso, está descrita bajo una máquina de estados que compara año, mes, día, hora, minuto y segundo entre los obtenidos por el GPS y los fijados por el usuario guardados en registros, permitiendo que el ancho de pulso también sea programable.

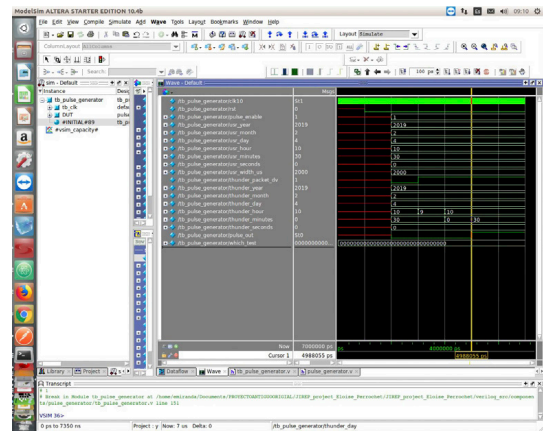


Figura 6. Simulación del Módulo Generador de Pulso

3.2. Pruebas con analizador lógico

Al enviar las configuraciones de inicio de pulso en formato UTC, el ancho de pulso y periodo de repetición, usando el analizador lógico se muestrea el pin de salida y se comprueba que genera los pulsos de salida deseados comprobando con un reloj UTC de internet.

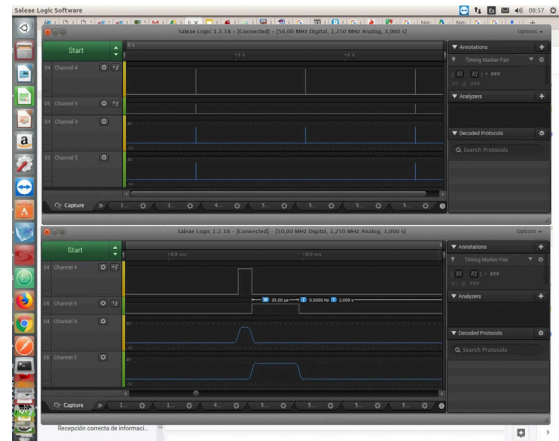


Figura 7. Vista del Pulso Generado con ayuda de Analizador Lógico Saleae

De igual forma se comprobó la generación de PPS divididos en base a las configuraciones hechas, en este caso con un factor 2, onda periódica, con 35us de ancho de pulso, 10 μs de desfase de la señal PPS original de tipo periódica.

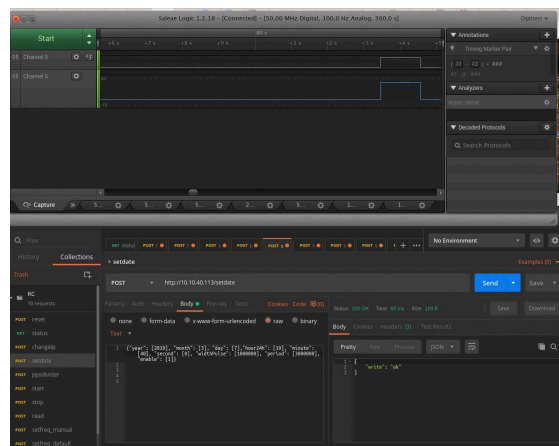


Figura 8. Vista del PPS y PPS dividido con las características de configuración desde la computadora

4. DISCUSIÓN

Lo fundamental para una programación y configuración adecuada tanto del MCU y FPGA ha sido trabajar directamente con los IDEs original del producto (Energia en el caso del MCU e ICEcube2 en el caso del FPGA), ambas soportan trabajar con distintos módulos que permiten una descripción y reutilización de código más adecuada. Con base al diseño de hardware, es preciso identificar aquellos pines con mayor interferencia o crosstalk producido por canales contiguos, de esta manera escoger aquellos intercaladamente resulta razonable. Con propósitos de que este hardware (PCB) pueda ser replicado en distintos radares VIPIR de la red LISN para que pueda más adelante implementarse un sistema multi estático, es conveniente que sea compatible para futuras mejoras y resistente a futuras fallas; por tal motivo se colocaron dos salidas de divisor de PPS y un backchannel UART entre MCU-FPGA en caso se presenten problemas con el SPI.

5. CONCLUSIONES

El uso de testbench (entornos de simulación) y un analizador lógico resultaron ser muy importantes a la hora de realizar las evaluaciones de funcionamiento. Por un lado, los testbench para verificar módulo por módulo y realizar una inspección visual de comportamiento de señales simuladas. Por otro lado, el analizador lógico Saleae permitió en diferentes etapas poder observar las tramas de envío de datos usando distintos analizadores para SPI y UART; así como su función tipo osciloscopio para observar el PPS y el pulso de salida. Se ha comprobado el funcionamiento completo y correcto del generador de pulso y divisor de PPS.

6. RECOMENDACIONES

Se puede implementar una interfaz web o un script de comandos que facilite al operario el acceso a las configuraciones del sistema

Se puede aprovechar la comunicación SPI o el back channel UART entre MCU y FPGA para acceder a la hora actual obtenida del Trimble.

Leer bien el manual del TinyFPGA Bx para conocer el procedimiento para cargarle programas con la herramienta Ice Cube y el "tinyprog" programmer para Linux.

Se puede parametrizar el módulo de divisor de PPS para aprovechar la segunda salida con conector BNC y crear un segundo PPS dividido bajo las configuraciones del operario.

7. AGRADECIMIENTOS

Al Instituto Geofísico del Perú por brindarme la oportunidad para desarrollar este proyecto que ha contribuido enormemente en mi desarrollo profesional, permitiéndome realizar un proyecto de ingeniería en todas sus etapas desde la identificación de ideas y objetivos, el diseño, análisis de resultados, implementación, ejecución y revisiones finales.

Igualmente deseo expresar mi gratitud al equipo de CIELO que me aceptó para desarrollar este sistema, brindándome su apoyo y disposición en lo posible para entender el funcionamiento de VIPIR. También agradecer al equipo de IDI, cuyo soporte fue imprescindible para el proyecto en sus diferentes etapas.

8. BIBLIOGRAFÍA

Thomas, Donald y Morby, Phil (2001). The Verilog Hardware Description Language.

TinyFPGA BX User Guide. Consulta: Febrero 2019. Disponible en:

<https://tinyfpga.com/bx/guide.html>

Tiva C Series TM4C129ENCPTD Microcontroller Data Sheet (Rev. B). Consulta: Febrero 2019. Disponible en:

<http://www.ti.com/tool/EK-TM4C129EXL>

ThunderBolt E User Guide. Consulta: Febrero 2019. Disponible en:

<http://trl.trimble.com/docushare/dsweb/Get/Document-383329/>

SISTEMA EN FPGA PARA PROPORCIONAR MARCAS DE TIEMPO BASADAS EN GPS A SISTEMAS DE ADQUISICIÓN EN EL RADIO OBSERVATORIO DE JICAMARCA

** FPGA system to provide GPS based timestamps for acquisition systems at the Jicamarca Radio Observatory*

Víctor Eduardo Vásquez Ortiz

Universidad Nacional Pedro Ruiz Gallo
Facultad de Ciencias Físicas y Matemáticas
victor.vasquez.1507@gmail.com

Asesor: Dr. Marco Milla Bravo

Radio Observatorio de Jicamarca - IGP

RESUMEN

En este trabajo se expone el diseño e implementación de un prototipo del sistema GTS (por sus siglas en inglés: GPS Timing System), el cual responde a la necesidad del Radio Observatorio de Jicamarca de cronometrar los eventos registrados por los radares de sus instalaciones. El sistema se desarrolló en un FPGA usando el lenguaje VHDL con un diseño estructural; permite extraer información de tiempo de un receptor GPS Trimble, con la que forma un reloj UTC con una precisión de hasta 100ns para poder cronometrar una señal de excitación externa (en este caso: la señal de transmisión de pulso de radar). Las marcas asignadas son almacenadas en registros dentro del módulo y serán transmitidas por un bus SPI.

Las pruebas arrojaron resultados esperados donde se ve una coherencia entre el periodo de las marcas de tiempo y la frecuencia del pulso de transmisión de radar. Por lo tanto, se concluye que el sistema es capaz de proporcionar estas marcas con una precisión de 100 ns.

Palabras clave: cronometraje, GPS, marca de tiempo, sincronización, FPGA

ABSTRACT

This work presents the design and implementation of a prototype of the GTS (GPS Timing System), which responds to the Jicamarca Radio Observatory necessity of timing the events registered by the radars on its facilities. The system was developed on a FPGA using VHDL language under a structural design; it allows to extract time information from a GPS Trimble receiver, with which creates a 100ns precision UTC clock to timestamp an external excitation signal (in this case: the radar transmission pulse). The timestamps are stored in registers inside the module and will be transmitted through a SPI bus.

The test showed expected results where there is coherence between the period of the timestamps and the radar transmission pulse. Hence, the system is capable of providing these timestamps with a 100 ns precision.

Keywords: timing, GPS, timestamp, synchronization, FPGA

1. INTRODUCCIÓN

Desde su creación en 1961, el Radio Observatorio de Jicamarca (ROJ) ha estado a la vanguardia en los estudios sobre la atmósfera, contando con una ubicación única, al estar situado en el ecuador magnético, y con uno de los radares más grandes y potentes del mundo.

A lo largo de los años, con desarrollo de hardware y software propio, el ROJ ha producido una gran cantidad de artículos científicos además de haber ofrecido los datos obtenidos por sus radares a la comunidad científica. Es por esta razón que, gracias al afán de seguir mejorando la calidad de la información brindada, se llevó a cabo este proyecto, el cual desarrolla el sistema GTS que es capaz de cronometrar a una escala precisa y de uso global cada evento registrado en el ROJ.

1.1. Planteamiento del problema

Actualmente, las señales obtenidas por los radares del ROJ pasan a través de un sistema de adquisición para luego ser enviadas como datos a una computadora, la cual les asigna una marca de tiempo.

El problema radica en que:

- El tiempo asignado a los datos corresponde al tiempo de llegada a la computadora y no al tiempo de adquisición.
- Sincronizar el tiempo de la computadora con una escala de tiempo estándar a través de internet no es una opción, pues la máxima precisión que se puede alcanzar va desde 1ms a 1s (Microsoft, 2017; Włodarczyk, 2013), sin contar los ajustes y saltos que se vuelven necesarios cuando el reloj local se desincroniza.
- La resolución normal que ofrece la computadora es en milisegundos y como en muchos experimentos realizados en el ROJ los periodos entre cada pulso de transmisión del radar están en el orden de unos pocos milisegundos (2.5 ms por ejemplo), se necesita una escala con una resolución mayor.
- El periodo transcurrido entre dos marcas de tiempo consecutivas cualesquiera no es el mismo pues la frecuencia del cristal de cuarzo dentro del reloj de tiempo real (RTC) de la computadora no es estable.
- Los datos obtenidos no se pueden comparar con los de otras estaciones pues no se cuenta con una escala de referencia.

Ante esta situación, se debía utilizar un reloj capaz de proporcionar el tiempo en una escala estándar y con una precisión de microsegundos, para que un sistema de cronometraje pueda extraer esta información y provea marcas de tiempo a los datos obtenidos por los radares.

Como en el ROJ se cuenta con receptores GPS que alcanzan una precisión de hasta 15ns y ofrecen el tiempo en las escalas GPS o UTC, desde el principio se decidió que estos equipos serían la fuente de tiempo y la escala estándar elegida debería ser UTC pues es la escala establecida a nivel global.

Por lo tanto, se decidió diseñar e implementar el sistema de cronometraje en un FPGA pues este dispositivo ofrece una gran versatilidad en el control y manejo de señales externas, capacidad de sincronismo y comunicación entre múltiples procesos trabajando en paralelo, y una arquitectura que permite trabajar a altas frecuencias.

1.2. Objetivos

1.2.1. Objetivo General

Diseñar e implementar un sistema en FPGA para proporcionar

marcas de tiempo basadas en GPS a sistemas de adquisición de radar.

1.2.2. Objetivos Específicos

- Diseñar un sistema capaz de proporcionar una marca de tiempo real, con una precisión en el orden de microsegundos, a una señal de control externa.
- Implementar dicho sistema en FPGA y simular su operación.
- Realizar pruebas reales y verificar el comportamiento del sistema en conjunto con los equipos de adquisición del ROJ.

1.3. Antecedentes

Abeysekara et al., (2014) describen el diseño y desempeño del sistema de Sincronización y Control GPS (GTC) del Observatorio de Rayos Gamma HAWC. El sistema, implementado en FPGA, proporciona marcas de tiempo totalmente sincronizadas con GPS con una precisión menor a 1us para cada evento registrado en HAWC, estas marcas son introducidas al sistema de adquisición de datos principal inmediatamente después de la etapa de recepción para evitar desfases con los datos registrados. Fue determinado que la precisión de las marcas de tiempo producidas por el sistema tenía un límite máximo de 25ns, que era más que suficiente para la precisión de 1us requerida por el observatorio.

Włodarczyk et al., (2013) describen un sistema de adquisición de datos con marcas de tiempo asignadas automáticamente, donde el tiempo es proporcionado por el receptor GPS: "Resolution T" del fabricante Trimble. El sistema está basado en un microcontrolador y logra una precisión con respecto al tiempo GPS o UTC mejor que $\pm 0.2\mu s$ con un retraso constante de aproximadamente 1.5us.

2. METODOLOGÍA

Para lograr los objetivos de la investigación fue necesario realizar las siguientes actividades:

- Se diseñó un circuito VHDL para obtener fecha y hora exacta UTC a partir del receptor GPS.
- Se diseñó un circuito VHDL de reloj en tiempo real con una resolución de microsegundos.
- Se diseñó un circuito VHDL que utilice los datos de fecha y hora UTC y el reloj en microsegundos para generar una marca de tiempo sincronizada con una señal de control externa, que sería el pulso de transmisión del radar proporcionado por el Controlador de Radar del ROJ.
- Se seleccionó una interfaz de transmisión de datos digital y se diseñó un circuito VHDL que usando dicha interfaz sea capaz de proporcionar la marca de tiempo a los sistemas de adquisición.
- Se probó los circuitos diseñados usando plataformas de simulación.
- Se integraron los circuitos para configurar el sistema que cumpla con el objetivo de la investigación.
- Se probó el sistema en conjunto con los equipos de adquisición del Radio Observatorio de Jicamarca.

El diagrama de la solución planteada se muestra en la Fig. 1.

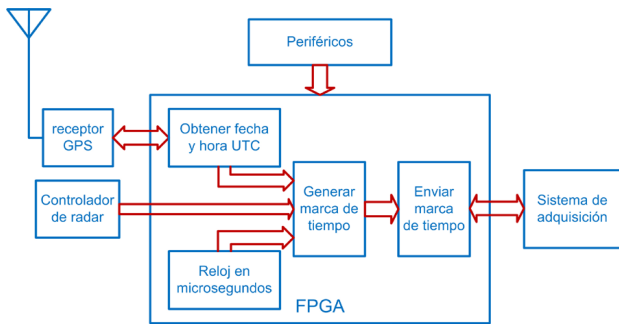


Figura 1. Modelo metodológico para el diseño e implementación del sistema GTS.

3. RESULTADOS

El sistema GTS desarrollado es un módulo hardware el cual extrae información de tiempo de un receptor GPS Thunderbolt E, a través de una interfaz RS-232, con dicha información forma un reloj UTC con una precisión inferior a 1us. Este sistema detecta una señal de excitación a la cual le asigna una marca de tiempo de 8 bytes dividida en 3 secciones:

- UNIX: (4 bytes) representación del tiempo UTC, elegida por ser la manera más fácil y corta de comunicar la información de tiempo con el sistema de adquisición del ROj.
- ms: (2 bytes) comunica el número de milisegundos en los que llegó la señal de excitación.
- us: (2 bytes) comunica el número de microsegundos en los que llegó la señal de excitación.

Estas marcas pueden ser solicitadas por un bus SPI esclavo, el que también permite configurar el sistema GTS.

El sistema se diseñó en VHDL usando una arquitectura estructural formada por varios componentes, por lo que también puede ser usado como parte de un sistema de adquisición más complejo. Para comprobar su correcto funcionamiento se realizarán pruebas de dos tipos.

La Fig. 2 muestra los componentes del sistema y la comunicación entre ellos.

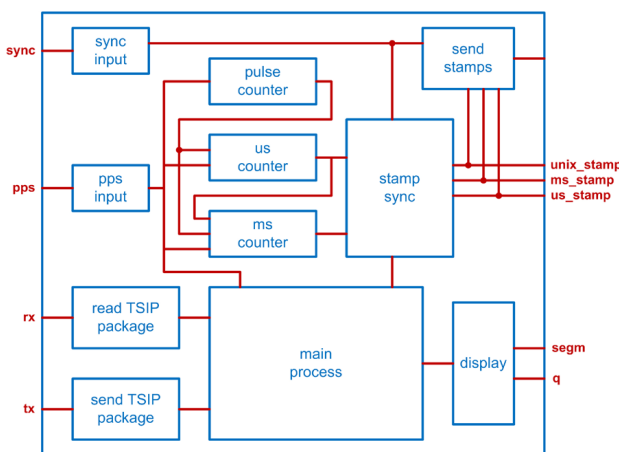


Figura 2. Componentes del sistema GTS.

3.1. Diseño

3.1.1. Componente de sincronización de señales externas

El componente *metastable_input* se diseñó para pasar una señal externa al dominio del reloj con el que funciona el sistema. Este

componente soluciona el problema de metaestabilidad¹ haciendo pasar la señal de *input* a través de un circuito que consiste en dos Flip-Flop tipo D ofreciendo una salida *output* que tiene el registro de los dos últimos estados de la señal *input* captados en el flanco de subida del reloj *clk*.

Este bloque se utiliza dos veces dentro del sistema, y toma las señales *pps* y *sync*, ver Figura 3.

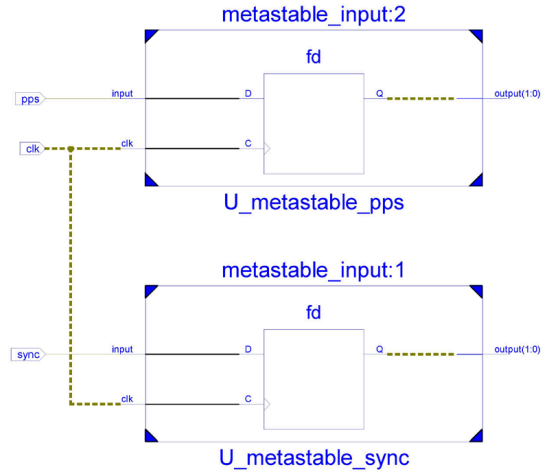


Figura 3. Circuito de bloques de sincronización.

3.1.4. Componente de comunicación serial UART

El componente *UART* fue implementado para establecer la comunicación serial con el receptor GPS, para esto se modificó ampliamente un módulo de *opencores.com* el cual consiste en dos máquinas de estado usadas para la recepción y para la transmisión de datos. La recepción cuenta con 4 procesos, mientras que la transmisión solo con 2 (Fig. 4).

La lógica de la parte de transmisión del componente se basa en la máquina de estados *tx_fsm* (Fig. 5) dentro del proceso *tx_process*, que se actualiza cada vez que el reloj *tx_clk* generado dentro de *tx_clk_generator* es "1", debido a esto el estado en la línea *tx* está retrasado un periodo respecto a *tx_fsm*.

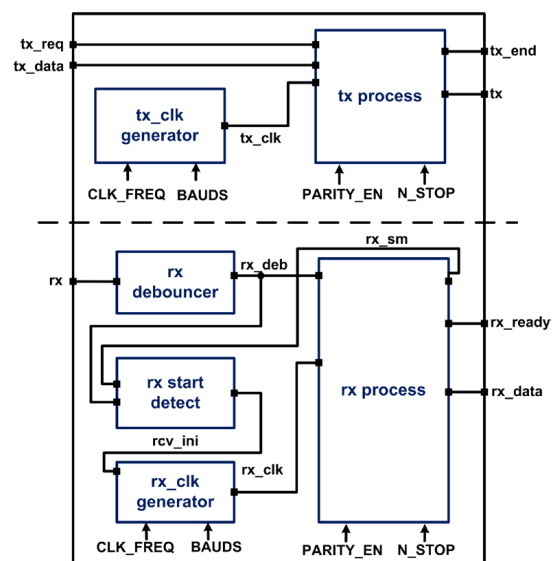


Figura 4. Diagrama de componente UART.

¹ Estado de la salida de un FF que oscila entre alto y bajo debido a una violación en el tiempo de establecimiento o mantenimiento de registro.

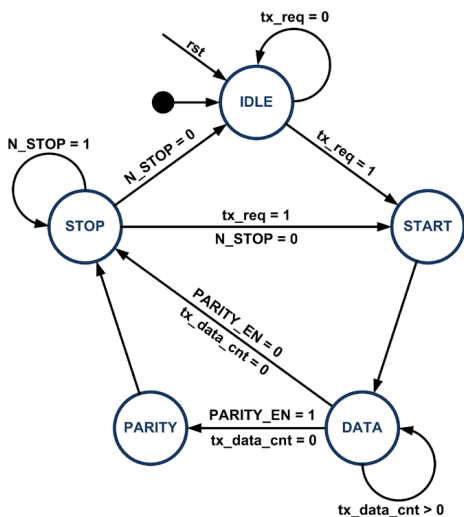


Figura 5. Máquina de estado de tx_process.

El comportamiento de la parte de recepción depende de 4 procesos (Fig. 4), el filtro antirebote del proceso rx_debouncer evita falsos cambios de estado en la señal de entrada externa rx ofreciendo la señal filtrada rx_deb, el proceso rx_start_detect detecta un flanco de bajada en rx_deb y si la máquina de estado rx_fsm se encuentra en idle o stop activa la señal interna rx_rcv_init que resetea a la mitad el contador que maneja la señal rx_clk

dentro del proceso rx_clk_generator asegurándose de que se adquiera el estado lógico de rx_deb a la mitad del periodo del bit (baudio), es por esto que rx_fsm está retrasado medio periodo con respecto a la línea rx_deb.

Cuando rx_clk es "1", cada vez que la cuenta llega a su máximo valor, se produce un cambio en el estado rx_fsm y se lee el valor de la línea rx_deb, así se van detectando los bits de inicio, datos y parada de acuerdo a la máquina de estado rx_fsm. Al recibir el último bit de parada se activará la señal rx_ready para indicarle al usuario que los datos almacenados en rx_data están listos para leerse. De detectarse una transición de alto a bajo cuando rx_fsm está en stop significa que una nueva transferencia está ocurriendo y rx_fsm pasa a start, caso contrario rx_fsm pasa al estado de reposo (idle).

3.1.5. Componentes contadores

En el sistema se utilizan 3 contadores basados en la señal clk: count_pulses, count_microseconds y count_milliseconds, los cuales dependen de la frecuencia del reloj CLK_FRQ (en MHz). Un diagrama completo de las conexiones entre los 3 contadores, así como sus entradas y salidas se puede apreciar en la Fig. 6.

El componente count_pulses consiste en un contador simple c de rango 0 a CLK_FREQ-1 que se incrementa a cada flanco de subida de clk y se reinicia cuando llega a CLK_FREQ-1 o cuando se detecta una transición de 0 a 1 en la señal pps_r. En la implementación final del sistema CLK_FREQ es 60, pues a esa frecuencia opera el Controlador de Radar del ROJ.

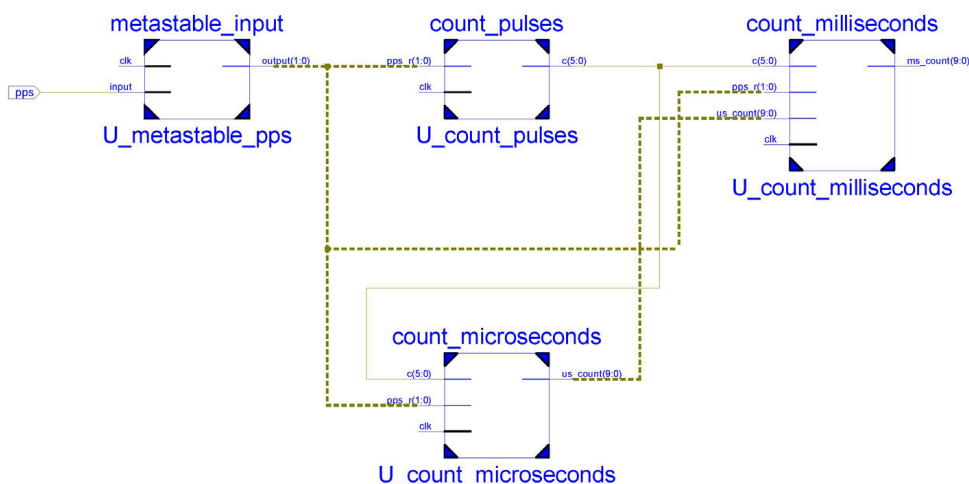


Figura 6. Circuito de contadores.

El componente count_microseconds formado por un contador us_count de rango 0 a 999, el cual se incrementa hasta llegar a 999 cuando el flanco de subida de clk coincide con c = CLK_FREQ-1 y solo se reiniciará cuando detecte una transición de 0 a 1 en la señal pps_r.

El componente count_milliseconds formado por un contador ms_count de rango 0 a 999, el cual se incrementa hasta llegar a 999 cuando el flanco de subida de clk coincide con c = CLK_FREQ-1 y con us_count = 999 y solo se reiniciará cuando detecte una transición de 0 a 1 en la señal pps_r.

3.1.6. Componente principal

El componente main es la sección principal del sistema y consiste en una máquina de estado mostrada en la Fig. 7, la cual espera por un primer pulso PPS del receptor GPS en el estado wait_for_pps para activarse y pasa a esperar los paquetes de datos enviados por el receptor (wait_for_package) que pasan a través del componente UART. La primera vez que se detecta un nuevo paquete, se

evalúa que tipo de data ha llegado en evaluate_package, de ser el paquete AB (el cual contiene la información de tiempo de GPS), pasa al estado compute_time donde calcula el tiempo UTC y llena los registros donde se almacena, de no serlo, se pasará al estado request_package_AB el cual se encargará de controlar el componente UART para solicitar al receptor ThunderBolt E el envío del paquete AB, una vez recibido recién se pasará al estado compute_time. Este ciclo se repetirá todo el día hasta el mediodía cuando se solicitará el envío del paquete AC el cual contiene información sobre los segundos intercalares² aplicados a la escala UTC, de esta manera, se aplican cambios al registro que almacena la marca UNIX parando o saltando un segundo a la cuenta. Luego de esto se seguirá en su funcionamiento normal.

² Segundos añadidos o quitados a la escala UTC para mantenerla en un desfase menor a 0.9s con respecto al tiempo UT1.

3.1.7. Componente de cronometraje

Este componente se encarga de recibir a la señal de excitación externa y asignarle el valor de tiempo actual, que es el valor de los contadores de microsegundos y milisegundos, así como el contador UNIX. La implementación final comunicará las marcas de este componente a través de un bus SPI el cual también podrá configurar el sistema GTS a través de un componente de registros. Para el primer prototipo se reusó el bloque serial UART para comunicar estas marcas.

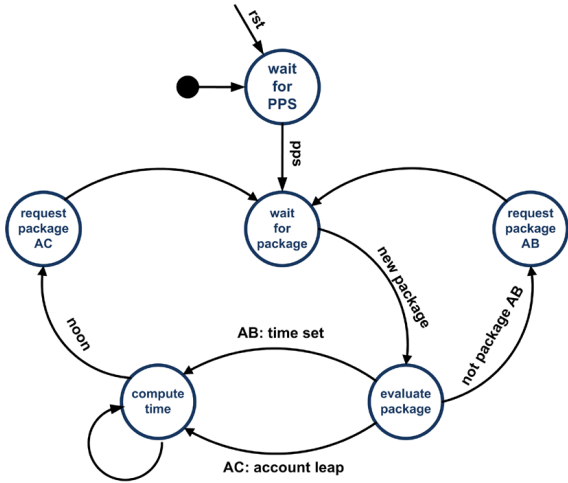


Figura 7. Máquina de estado de componente principal.

3.2. Pruebas

Hasta el momento se ha realizado un tipo de prueba para probar el funcionamiento correcto del sistema y observar las diferencias entre cada marca de tiempo que deben concordar con la frecuencia establecida para la prueba. Se pretende realizar pruebas de portabilidad implementando el sistema en dos tarjetas FPGA y ver su respuesta. Un extracto de las marcas obtenidas con una señal de excitación de 20Hz (salida del controlador de radar) se puede ver en la Tabla 1.

UNIX	ms	us
1521667558	750	12
1521667558	800	12
1521667558	850	12
1521667558	900	12
1521667558	950	12
1521667559	0	12
1521667559	50	12
1521667559	100	12

Tabla 1. Marcas de tiempo obtenidas el 21 de Marzo del 2018

4. CONCLUSIONES

Se implementó el primer prototipo del sistema GTS capaz de proporcionar marcas de tiempo UTC con una precisión de 0.1us.

El sistema funcionó correctamente en las pruebas en conjunto con los equipos del ROJ.

El sistema puede usarse como módulo hardware dentro de un sistema de adquisición.

El sistema GTS puede ser usado en cualquier otra aplicación donde se necesite marcar una señal de excitación.

5. AGRADECIMIENTOS

A Marco Milla, Joaquín Verástegui y Danny Scipión del Radio Observatorio de Jicamarca por su apoyo y guía en el desarrollo de esta tesis.

6. BIBLIOGRAFÍA

Abeysekara et al. (2014) Abeysekara, A. U., Ukwatta, T. N., Edmunds, D., Linnemann, J., Imran, A., Kunde, G., y Wisher, I. (2014). GPS Timing and Control system of the HAWC detector.

Microsoft, (2017). Support boundary to configure the Windows Time service for high-accuracy environments. Microsoft Support. Article ID: 939322.

Włodarczyk et al. (2013) Włodarczyk, P., Pustelny, S., Budker, D., y Lipinski, M. Multi-channel data acquisition system with absolute time synchronization. Nuclear Instruments and Methods in Physics Research Section A.

VARIABILIDAD DE ESTRELLAS RÁPIDAMENTE OSCILANTES AP: HD 217522

**Variability of rapidly oscillating Ap stars: HD 217522*

Raúl Santa Cruz

Universidad Nacional de Ingeniería
Facultad de Ciencias – Escuela Profesional de Física
raulsantacruz0@gmail.com

Asesor: Dr. Antonio Pereyra

Unidad Funcional de Astronomía -
Observatorio de Huancayo - IGP

RESUMEN

Se analizó la curva de luz de la estrella rápidamente oscilante Ap HD 217522 de modo a confirmar sus periodos típicos de variabilidad. La fotometría fue realizada usando el telescopio OAUNI instalado en el Observatorio de Huancayo. Luego de un filtrado apropiado de valores extremos en la curva de luz, el análisis utilizando periodogramas con el algoritmo Lomb-Scargle detectó el periodo principal de oscilación $P1 = 14.43 \pm 0.74$ min. El ajuste del diagrama de fase respectivo permitió determinar su amplitud media de oscilación (~ 11 mmag). Adicionalmente, el segundo periodo también fue detectado, $P2 = 7.63 \pm 0.46$ min.

Palabras clave: fotometría, variabilidad estelar, estrellas roAp, curva de luz, periodograma

ABSTRACT

This work analyzes the light curve of the rapidly oscillating Ap star HD 217522 in order to confirm its typical variability periods. The photometry was performed using the OAUNI telescope installed at the Huancayo Observatory. After removing outliers on the light curve, the analysis using periodograms with the Lomb-Scargle algorithm detected the main period of oscillation $P1 = 14.43 \pm 0.74$ min. A fitting of the associated phase diagram let to determine the mean amplitude of oscillation (~ 11 mmag). In addition, the second period was also detected, $P2 = 7.63 \pm 0.46$ min.

Keywords: photometry, stellar variability, roAp stars, light curve, periodogram

1. INTRODUCCIÓN

Estrellas variables pulsantes muestran variaciones periódicas de sus capas superficiales en expansión y contracción (Kartunen 2007). La pulsación es radial cuando la estrella mantiene su forma esférica y no radial cuando existe una desviación periódica en su esfericidad. Los tipos de variables pulsantes pueden ser clasificadas por el periodo de pulsación y por la forma de sus curvas de luz. Estas últimas son el resultado de la masa y del estado evolutivo de la propia estrella. Cuando el periodo es del orden de pocas horas a decenas de días, la estrella es llamada de un pulsador de periodo corto. Este grupo incluye las estrellas Cefeidas, las RR Lyrae y las delta Scuti, las cuales ocupan la banda de inestabilidad en el diagrama Hertzsprung-Russell (Catelan 2015).

Las estrellas rápidamente oscilantes Ap (roAp) también están situadas sobre la banda de inestabilidad, cercanas a la posición de las estrellas delta Scuti en el diagrama HR. Las estrellas roAp son estrellas de tipo A con fuertes campos magnéticos y composición química superficial peculiar. Se conocen alrededor de cuarenta estrellas roAp con variabilidad fotométrica del orden de centésimas de magnitud y periodos debajo de veinte minutos. El modelo del pulsador oblicuo es usado para explicar los múltiples modos de oscilación observados en las estrellas roAp. En dicho modelo es necesario que el eje magnético este desalineado con respecto del eje de rotación estelar (Kurtz 1982, Martínez et al. 1998)

HD 217522 es una estrella roAp de magnitud visual $V = 7.52$ mag, cuyas observaciones fotométricas iniciales revelaron una frecuencia de pulsación $F_1 = 1.21509$ mHz (o periodo $P_1 = 13.72$ min, Kurtz 1983) con amplitudes de oscilación del orden de 4 mmag. Adicionales observaciones encontraron una bien definida segunda frecuencia $F_2 = 2.0174$ mHz (o periodo $P_2 = 8.3$ min, Kreidl et al 1991). De manera interesante, HD 217522 muestra amplitudes de modulación sobre una escala de tiempo del orden de un día, que son mucho más cortas de lo que se ha observado en otras estrellas (Medupe et al. 2015).

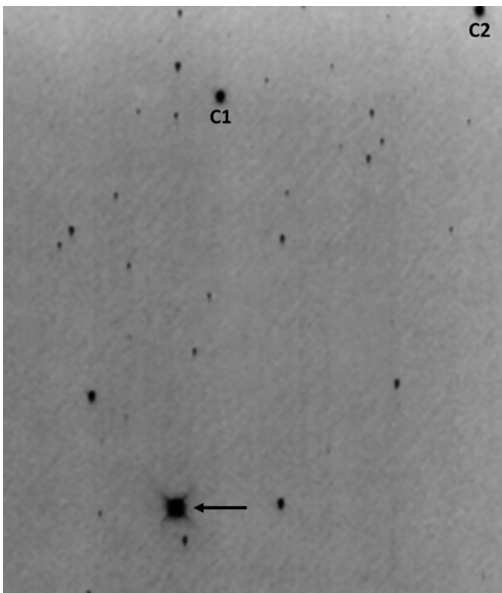


Figura 1: Campo de la estrella HD 217522 observada el 30/07/2017 usando filtro V. Se muestra una sección de $\sim 13' \times 15'$. La imagen es el resultado del apilamiento de $450 \text{ frames} \times 2s = 1.1$ horas (incluido el tiempo muerto). Norte es arriba y Este a la izquierda. HD 217522 está indicada (flecha), así como las estrellas de comparación C1 (HD 217485, $V_{\text{mag}} = 9.70$) y C2 (HD 217396, $V_{\text{mag}} = 9.34$).

2. OBSERVACIONES Y REDUCCIÓN DE DATOS

Las observaciones fueron realizadas usando el telescopio OAUNI de 0.5m de apertura (Pereyra et al. 2015) instalado en la sede de Huancayo del IGP. Las observaciones de la estrella HD 217522 fueron realizadas la noche del 30/07/2017 durante la estación de seca en los Andes centrales peruanos. El detector usado fue un CCD de iluminación frontal modelo STXL-6303E y manufacturado por SBIG. El área del CCD cuenta con 3072×2048 píxeles con un tamaño de $9\mu\text{m}$ por píxel. El detector junto con la razón focal $f/8.2$ del sistema óptico proporcionan una escala de placa de $0.45''/\text{píxel}$ y un campo de visión total de $23' \times 15'$. Todas las observaciones fueron realizadas en el filtro V.

La Figura 1 muestra el campo observado alrededor de la estrella HD 217522. Se adquirió una de secuencia 450 imágenes consecutivas de veinte segundo de integración por imagen individual. Considerando el tiempo muerto, el objeto fue monitoreado alrededor de 1.1 hora, tiempo suficiente para obtener alguno de los periodos encontrados en la literatura para esta estrella.

Todas las imágenes fueron reducidas usando el ambiente IRAF con las típicas correcciones de corriente oscuro y campo plano. Se utilizaron rutinas IRAF desarrolladas especialmente para sistematizar el proceso de alineamiento de imágenes. Finalmente fue utilizado DAOPHOT para la realización de fotometría de apertura. Se escogieron radios de apertura fotométrica de dos (r_2) y tres (r_3) veces el valor promedio del seeing medido y se escogió el que minimizó el error fotométrico medio. Mejores resultados se consiguieron con el radio de apertura r_2 .

Se utilizó la técnica de fotometría diferencial para construir las curvas de luz mostradas en este trabajo. Así, la magnitud instrumental del objeto de interés fue descontado de la magnitud instrumental de una estrella de comparación en el mismo campo de visión. Una segunda estrella de comparación fue siempre medida de modo a hacer un doble chequeo. El amplio campo de visión de nuestro detector facilita la elección de la mejor estrella de comparación en cada caso. En este caso, la estrella de comparación C2 (ver Figura 1) fue la que mejor resultados produjo en el análisis a seguir.

3. RESULTADOS

La Figura 2a muestra la curva de luz de la estrella HD 217522. Es interesante notar que el error fotométrico promedio en nuestra fotometría diferencial (3 mmag) por medida individual es al menos un orden de magnitud menor que la típica dispersión pico a pico observada (28 mmag). Esta condición es importante para tener chance de encontrar periodos ocultos asociados a variaciones en amplitud en escalas de mmag usando periodogramas.

La Figura 2b muestra el periodograma asociado a la curva de luz de HD 217522 utilizando el algoritmo Lomb-Scargle (Scargle 1982). El periodo asociado al primer pico encontrado es $P_1 = 14.07 \pm 0.62$ min. Este valor está en excelente acuerdo con el pico fundamental asociado a HD 217522 encontrado previamente (Kurtz 1983). Por otro lado, el periodograma indica también un pico de baja amplitud asociado al segundo periodo encontrado en la literatura para esta estrella. Nuestro valor es $P_2 = 7.99 \pm 0.26$ min, también en buena concordancia con Kreidl et al (1991).

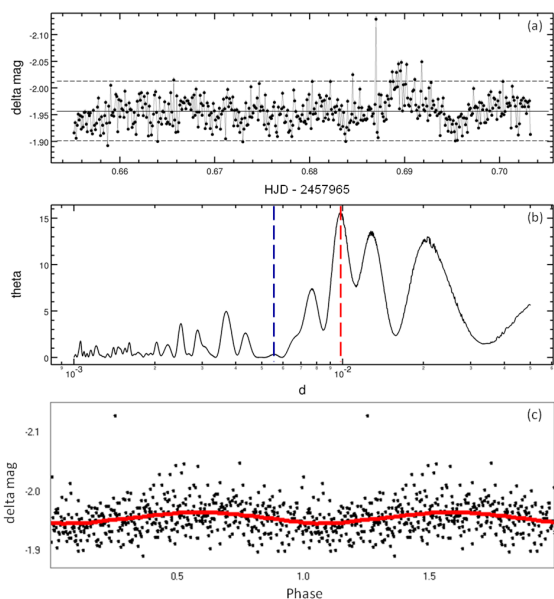


Figura 2: (a) Curva de luz de HD 217522. El valor promedio está indicado (línea sólida), así como la dispersión 2xsigma (línea cortada); (b) Periodograma. Se muestran los periodos (en días) encontrados P1 (rojo) y P2 (azul); (c) Diagrama de fase para P1. Un ajuste tipo spline (rojo) indica una amplitud de 19 mmag.

Desde que el periodo principal P_1 fue detectado sin problema en nuestro análisis de periodograma, la Figura 2c muestra el diagrama de fase correspondiente a este periodo. Para cuantificar la amplitud de la variación de P_1 se procedió a hacer un ajuste tipo spline al diagrama de fase, lo que resultó en una amplitud de 19 mmag para este periodo. Este valor es al menos un orden de magnitud mayor que detecciones previas (≤ 4 mmag, Kurtz 1983).

El diagrama de fase anterior muestra claramente varias medidas fotométricas que se apartan de la oscilación esperada. Esto puede deberse a eventuales variaciones en la calidad de cielo debido a nubes pasajeras, viento y/o un incremento de la turbulencia lo que empeora el seeing durante el intervalo medido. Para explorar estos factores se procedió a filtrar la curva de luz eliminando valores fotométricos extremos que se aparten en dos veces la dispersión típica. Esto está indicado en la Figura 2a donde se muestra el valor fotométrico promedio de la curva de luz, así como los intervalos máximo y mínimo del filtrado indicado (2xsigma).

La Figura 3a muestra la curva de luz filtrada por el ejercicio anterior, a la cual se le procedió nuevamente a calcularle su periodograma respectivo (Figura 3b). El nuevo periodograma muestra como era de esperar el pico principal P_1 , pero ahora el pico secundario P_2 se hace mucho más evidente, lo que indica que el filtrado realizado fue exitoso. Los valores encontrados para los periodos buscados fueron $P_1 = 14.43 \pm 0.74$ min y $P_2 = 7.63 \pm 0.46$ min. Como puede notarse los errores asociados a los periodos se incrementaron hasta casi un factor dos, especialmente en P_2 . Por otro lado, y a pesar de ser notorio ahora el segundo periodo P_2 , su valor nominal queda algo subestimado del valor publicado. Finalmente, se reajustó la amplitud de la oscilación usando el diagrama de fase para el periodo P_1 (Figura 3c), resultando en una amplitud de oscilación de 11 mmag, la cual es más cercana a los valores previamente medidos (Kurtz 1983), pero aún superior. La Tabla 1 resume los periodos encontrados con sus amplitudes de oscilación de modo a facilitar su comparación con los valores publicados.

Curva de luz	"P1 (min)"	"Amp1 (mmag)"	"P2 (min)"
cruda	14.07 ± 0.62	19	7.99 ± 0.26
filtrada ^a	14.43 ± 0.74	11	7.63 ± 0.46
ref.	13.72^b	$\leq 4.2^b$	8.3^c

^afiltrado a 2xsigma; ^b Kurtz (1983); ^c Kreidl et al (1991)

Tabla 1. Periodos de oscilación encontrados en HD 217522

4. DISCUSIÓN

Es interesante notar que la presente detección de los periodos principales de pulsación de la estrella roAp HD 217522 se ha realizado con una curva de luz que cubre un intervalo de monitoreo fotométrico de algo más de sesenta minutos. Esto es, el monitoreo ha cubierto casi cinco periodos P_1 y casi ocho periodos P_2 . Es de esperar que las precisiones en la determinación de los periodos mejoren substancialmente con rangos de monitoreo mayores.

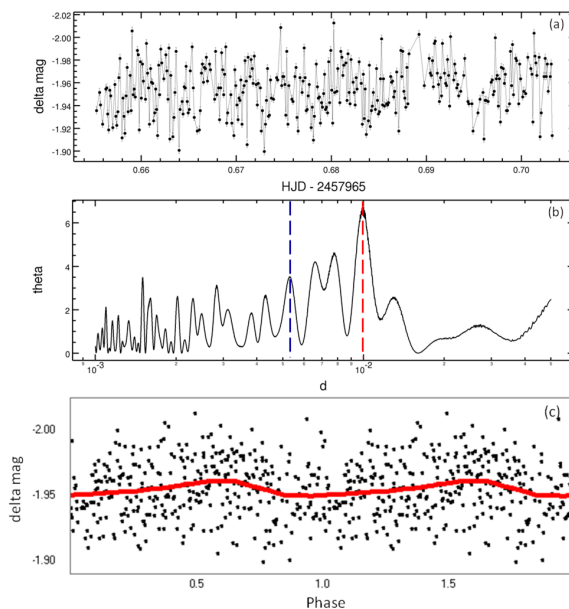


Figura 3: Similar a Fig. 2 para datos filtrados a 2xsigma. (c) Amplitud de ajuste spline = 11 mmag.

La amplitud de oscilación detectada para el periodo P_1 aparentemente está superestimada, si es comparada con los valores de Kurtz (1983). Esto puede reflejar el hecho que estamos cerca a nuestro límite de detección fotométrico, toda vez que nuestra precisión fotométrica por medida individual ya es del orden de mmag. Otra posibilidad es que realmente estemos detectando una variación mayor en la amplitud de oscilación, teniendo en consideración su variación en escala de días ya indicada por Medupe et al (1995) para el caso de HD 217522. Estas posibilidades serán exploradas en nuevas curvas de luz del objeto realizadas con un mayor rango temporal y ya observadas a la fecha.

5. CONCLUSIONES

El presente trabajo demuestra la viabilidad de analizar periodos de oscilación de estrellas roAp con el instrumental actual del proyecto OAUNI.

El análisis de la curva de luz de la estrella roAp HD 217522 a través de periodogramas mostró sus dos periodos de oscilación

característicos: $P_1 = 14.43 \pm 0.74$ min y $P_2 = 7.63 \pm 0.46$ min. Estos valores fueron obtenidos luego de filtrar apropiadamente la curva de luz evitando valores extremos asociados a turbulencia atmosférica.

El ajuste de la fotometría en el diagrama de fases para el primer periodo P_1 permitió determinar la amplitud de dicha oscilación. Ésta resultó ser de 11 mmag apenas ~ 4 veces nuestro límite de detección fotométrico.

Nuevos datos (ya recolectados) para HD 217522 permitirán mejorar las precisiones en los periodos encontrados al analizar rangos temporales mayores al utilizado en este trabajo.

6. AGRADECIMIENTOS

Agradecemos al Instituto Geofísico del Perú GP por la subvención económica otorgada para la realización de este trabajo. El proyecto OAUNI agradece el apoyo económico de The World Academy of Sciences (TWAS), del Rectorado y del Instituto General de Investigación (IGI) de la UNI, y de Concytec (Convenio 102-2015 Fondecyt). Somos gratos al personal del Observatorio de Huancayo por el apoyo logístico.

7. BIBLIOGRAFÍA

Catelan, M., Smith, H. A. (2015). Pulsating stars Wiley-VCH Verlag Ed., p 50

Karttunen, H., Kröger, P., Oja, H., Poutanen, M., Donner, K. J. (2007). Fundamental Astronomy (5th. Edition) Springer Ed., p 281

Kurtz, D. W. (1983). The discovery of 13.72-min oscillations in the cool magnetic AP star HD 217522, Monthly Notices of the Royal Astronomical Society, Vol. 205, 3-10, doi: 10.1093/mnras/205.1.3

Kreidl, T. J., Kurtz, D. W., Bus, S. J. et al. (1991). A frequency analysis of new observations of the rapidly oscillating AP star HD 217522: evidence for mode-switching, Monthly Notices of the Royal Astronomical Society, Vol. 250, 477, doi: 10.1093/mnras/250.2.477

Medupe, R., Kurtz, D. W., Elkin, V. G., Mguda, Z., Mathys, G. (2015). Monthly Notices of the Royal Astronomical Society, Volume 446, Issue 2, p.1347-1355, Monthly Notices of the Royal Astronomical Society, Vol. 446, Issue 2, p.1347-1355, doi: 10.1093/mnras/stu2189

Pereyra, A., Tello, J., Meza, E., Cori, W., Ricra, J., Zevallos, M. (2015), Observatorio Astronómico de la UNI - OAUNI: primera luz, Revista de la Facultad de Ciencias de la UNI, 18, 1, 4-9

Scargle, J. D. (1982). Studies in astronomical time series analysis. II. Statistical aspects of spectral analysis of unevenly spaced data, Astrophysical Journal, Vol. 263, p. 835-853, doi: 10.1086/160554



CIENCIAS DE LA ATMÓSFERA E HIDRÓSFERA

SECCIÓN 2

CARACTERIZACIÓN DE LA DINÁMICA OCEÁNICA DE LA COSTA PERUANA BAJO DIFERENTES ESCENARIOS DE ESFUERZO DE VIENTOS

**Characterization of the Peruvian coastal dynamics under different scenarios of wind stress*

Katerine Elsy Ticse De La Torre

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Físicas
Elsy50988@gmail.com

Asesora: Dra. Ivonne Montes

Sub Dirección de Ciencias de la Atmósfera e Hidrósfera - IGP

RESUMEN

La dinámica oceánica de la costa peruana se mantiene gracias a la configuración espacio-temporal del esfuerzo de vientos, que genera y modula procesos físicos importantes como el afloramiento costero. El presente trabajo realiza un análisis de sensibilidad asociado al aumento o disminución del esfuerzo de vientos a fin de analizar los cambios en la estructura de la columna de agua frente a la costa peruana. Para ello, se analizan tres escenarios numéricos que han sido forzadas con diferentes esfuerzos de vientos (mínimo, máximo, 5 veces) generados y modificados a partir de un experimento realista para el Pacífico Tropical Este. Los resultados muestran que los cambios más significativos ocurren sobre los ~550 m de profundidad, donde se evidencian cambios en la estructura térmica, intensidad y distribución de las corrientes oceánicas (por ejemplo, la Corriente Subsuperficial Perú-Chile (PCUC) muestra una relación directa con el estrés del viento). Respecto a la actividad de mesoescala, los cambios drásticos son respecto a sus características (Radio, Amplitud, Densidad de Energía (EI) e Índice de Actividad de remolino (EAI)); siendo el cambio más extremo en la EAI debido a la variación de energía cinética asociada.

Palabras clave: Esfuerzo de vientos, Costa Peruana, escenarios numéricos, Actividad de mesoescala.

ABSTRACT

Peruvian coastal dynamics are mostly driven by wind stress along the coast and nearshore wind stress curl, which generates and modulates important physical processes such as the coastal upwelling. We performed a sensitivity analysis associated to the increase or decrease of the wind stress to assess the regional impact the changes in the structure of the water column. For this, three numerical scenarios have been analyzed which are forced with different wind stress (maximum, minimum, quintuple) generated and modified from a realistic experiment for the Eastern Tropical Pacific. The results show that the most significant changes occur over ~550 m depth, where changes in the thermal structure, in intensity, and distribution of ocean currents are evidenced (for example, the Peru-Chile Undercurrent - PCUC - shows a direct relationship to wind stress). Respect to mesoscale activity, the drastic changes are related to the eddy characteristics (Radius, Amplitude, Energy Density (EI) and eddy activity index (EAI)); being the most extreme change in the EAI due to the associated kinetic energy variation.

Keywords: Wind stress, Peruvian coast, numerical scenarios, mesoscale activity

1. INTRODUCCIÓN

En la costa peruana los vientos alisios son los principales responsables de desarrollar y mantener la dinámica oceánica local (Albert et al., 2010; Dijkstra, 2013). Estos vientos, denominados alisios, soplan hacia el Ecuador siendo casi paralelos a la línea de costa, e incrementando su magnitud costa afuera provocando un ascenso de agua fría debido al proceso de afloramiento costero, desarrollado a través del mecanismo de Ekman (Gutiérrez et al., 2011; Kessler, 2006).

Otro mecanismo importante modulado por la acción del viento, en superficie, viene a ser la actividad de mesoescala o remolinos de mesoescala (Xu et al., 2016). Los remolinos de mesoescala frente a la costa peruana presentan bajos niveles ($200 - 300 \text{ cm}^2 \text{ s}^{-1}$) de energía cinética de remolino (EKE), sin embargo, contribuyen en los procesos de mezcla (Bakun et al., 2015; Brown et al., 1989; Stewart, 2008) transfiriendo grandes cantidades de calor, masa y energía, así como propiedades biogeoquímicas desde la costa hacia costa afuera (Chaigneau et al., 2008). Frente a Perú, la costa sur es identificada como área con mayor generación de remolinos, y frente a los 9°S (Chimbote) costa afuera como área con mayor frecuencia de remolinos (Chaigneau and Pizarro, 2005; Penven et al., 2005).

Debido a la importante acción de los vientos sobre la dinámica de la costa peruana, un cambio en su comportamiento podría repercutir en los diferentes procesos que se desarrollan con normalidad. Por ejemplo, Bakun (1990) planteó la hipótesis de una intensificación de vientos en un escenario futuro, señalando que ello intensificaría el afloramiento costero reflejado por una disminución de temperatura superficial del mar. Sin embargo, más adelante Goubanova et al. (2011) y Steiner and Sou (2015) indicaron todo lo contrario; que la estacionalidad y magnitud del esfuerzo de vientos frente a la costa peruana se debilitaría.

El presente trabajo emplea un modelo oceánico regional para realizar un análisis de sensibilidad asociado al aumento o disminución del esfuerzo de vientos a fin de analizar los cambios en la estructura de la columna de agua frente a la costa peruana. Los cambios en el esfuerzo de los vientos son seleccionados de acuerdo a las hipótesis de Bakun (1990), Goubanova et al. (2011) y Steiner and Sou (2015), y como base o experimento control se utiliza la configuración de Montes et al. (2010).

2. METODOLOGÍA

2.1. Configuraciones Climatológicas del modelo oceánico

El modelo numérico empleado es el modelo oceánico "ROMS" (Regional Ocean Modelling System), el cual representa la dinámica del océano de superficie a fondo, resolviendo las ecuaciones primitivas basadas en la aproximación de Boussinesq y el balance hidrostático (ver detalles en: Marchesiello et al., 2001; Shchepetkin and McWilliams, 2003). La simulación de control denominada RSoda fue desarrollada por Montes et al. (2010) bajo condiciones climatológicas. El dominio simulado se extiende entre los 70°W hasta 90°W y los 4°N hasta 20°S , teniendo resolución horizontal de $1/9^\circ$ ($\sim 12 \text{ km}$) y 32 niveles verticales. La topografía del modelo se obtuvo a partir de los datos de ETOPO2. Las condiciones de fronteras abiertas provienen de la climatología de datos del modelo global SODA (Simple Ocean Data Assimilation). Los forzantes superficiales de flujos de calor y agua fresca son extraídos a partir de la climatología mensual de COADS (Comprehensive Ocean - Atmosphere Data Set). El forzante superficial de esfuerzo de los vientos es extraído a partir del promedio climatológico de la base de datos de QuickSCAT (Satellite Scatterometer Data). El modelo es corrido por 9 años, donde los primeros 4 años son considerados como el spin-up. Las salidas numéricas del modelo son almacenadas como promedios de cada 3 días.

Para los experimentos numéricos se realizaron 3 simulaciones climatológicas adicionales, las cuales son idénticas al escenario de control (RSoda) excepto por el forzante de vientos, el cual fue modificado para cada escenario (ver Tabla 1). Dichas modificaciones se realizaron al forzante superficial utilizados para el experimento control, específicamente sobre el esfuerzo de viento zonal (τ_x) y meridional (τ_y). El primer y segundo experimento obedecieron al debilitamiento e intensificación del esfuerzo de los vientos, siendo denominados RSoda_Qmin y RSoda_Qmax, respectivamente. En dichos experimentos se consideran valores uniformes en el espacio y tiempo, donde RSoda_Qmin considera los valores mínimos de ' τ_x ' y ' τ_y ' entre la costa y 250 km costa afuera del promedio anual, mientras que, RSoda_Qmax utiliza el valor máximo encontrado. El tercer experimento intensifica el esfuerzo de los vientos 5 veces el valor original, siendo denominado como RSoda_Q5. Los modelos son corridos por 10 años, siendo almacenados en promedio cada tres días.

	Escenarios de Simulación			
	RSoda	RSoda_Qmin	RSoda_Qmax	RSoda_Q5
Condición de frontera	ETOPO2	ETOPO2	ETOPO2	ETOPO2
Forzante de viento	QuickSCAT	$T_x = -0.0023$ $T_y = 0.0092$	$T_x = -0.0344$ $T_y = 0.0516$	5*QuickSCAT
Forzante de aguas frescas y cálidas	COADS	COADS	COADS	COADS
Forzante de fronteras abiertas	SODA	SODA	SODA	SODA

Tabla 1: Condición de frontera y forzantes para cada simulación climatológica

Para este estudio se calcula los promedios anuales de la temperatura, salinidad y campo de velocidades de las salidas numéricas. Para el análisis de la actividad de remolinos se analizan los datos numéricos en promedio de cada tres días. Todos los análisis se realizan en el área entre los 6°S hasta 15°S y los 75°W hasta 85°W .

2.2. Algoritmo de detección de remolinos

De manera similar a la metodología usada por Yi et al. (2014), se elaboró un algoritmo para la detección de remolinos, en la cual se usan dos criterios de identificación. El primer criterio se basa en la metodología de detección física que consiste en la identificación del centro del remolino (Penven et al., 2005), y el segundo criterio se basa en la metodología de detección geométrica la cual identifica las fronteras de los remolinos (Chaigneau et al., 2008). Ambos criterios se unen bajo la siguiente premisa 'Se considera como remolino detectado al contorno que encierra uno o varios centros identificados' (ejem. Figura 1).

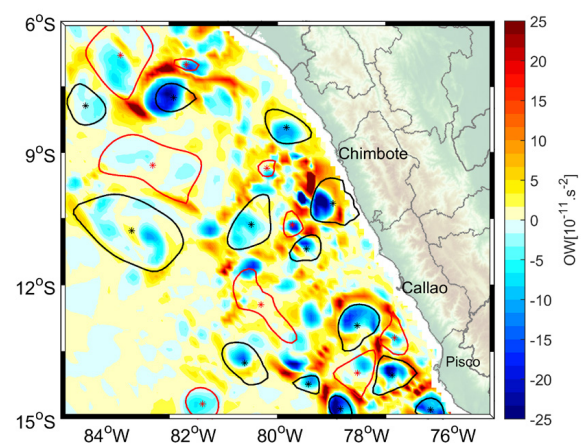


Figura 1. Parámetro Okubo Weiss [10^{-11} s^{-2}] en superficie del 12 de Junio del año 22° modelado, para RSoda. Los contornos negros y rojos indican los remolinos anticiclónicos y ciclónicos, respectivamente.

Además, el algoritmo calcula las características del remolino, como son el radio, amplitud, densidad de Energía e índice de la actividad de remolinos. Cabe resaltar que la resolución de los datos numéricos es de $1/9^\circ$ (~12km), por lo cual los contornos con áreas menores a 12.5 km no son considerados.

2.2.1. Centros de Remolinos

Para la identificación de los centros de remolinos se considera la proximidad o coincidencia de los centros de variaciones de la Anomalía del nivel del Mar (SLA) y el campo de velocidad de corriente (zonal - U y meridional -V). Además, estas variaciones deben estar dentro de un área con valor negativo del parámetro Okubo Weiss, ya que el valor negativo de este parámetro representa la vorticidad del fluido.

2.2.2. Bordes de Remolinos

Para la identificación de bordes de los remolinos se sigue la metodología de detección geométrica o 'Winding Angle' (WA). Esta metodología calcula la función de corriente utilizando las anomalías de velocidades de corriente (U' y V'), considerando al contorno cerrado más grande y con menor deformación como borde del remolino.

2.2.3. Características de Remolinos

De acuerdo a Chaigneau et al. (2008) se calculan:

- Radio Aparente del remolino (ER):

$$ER^2 = \sqrt{\frac{EA}{\pi}}$$

- Amplitud del remolino:

$$Amplitud = |SLA_{centro} - SLA_{borde}|$$

- Densidad de Energía (EI):

$$EI = \frac{EKE'''}{EA}$$

- Índice de Actividad de remolino (EAI):

$$EAI = \frac{NEI}{EA}$$

Donde 'EA' es el área aparente del remolino, 'EKE''' representa el promedio, 'EKE''' es la energía cinética encerrada en el remolino y 'N' es la cantidad de los remolinos detectados en un tiempo 't'.

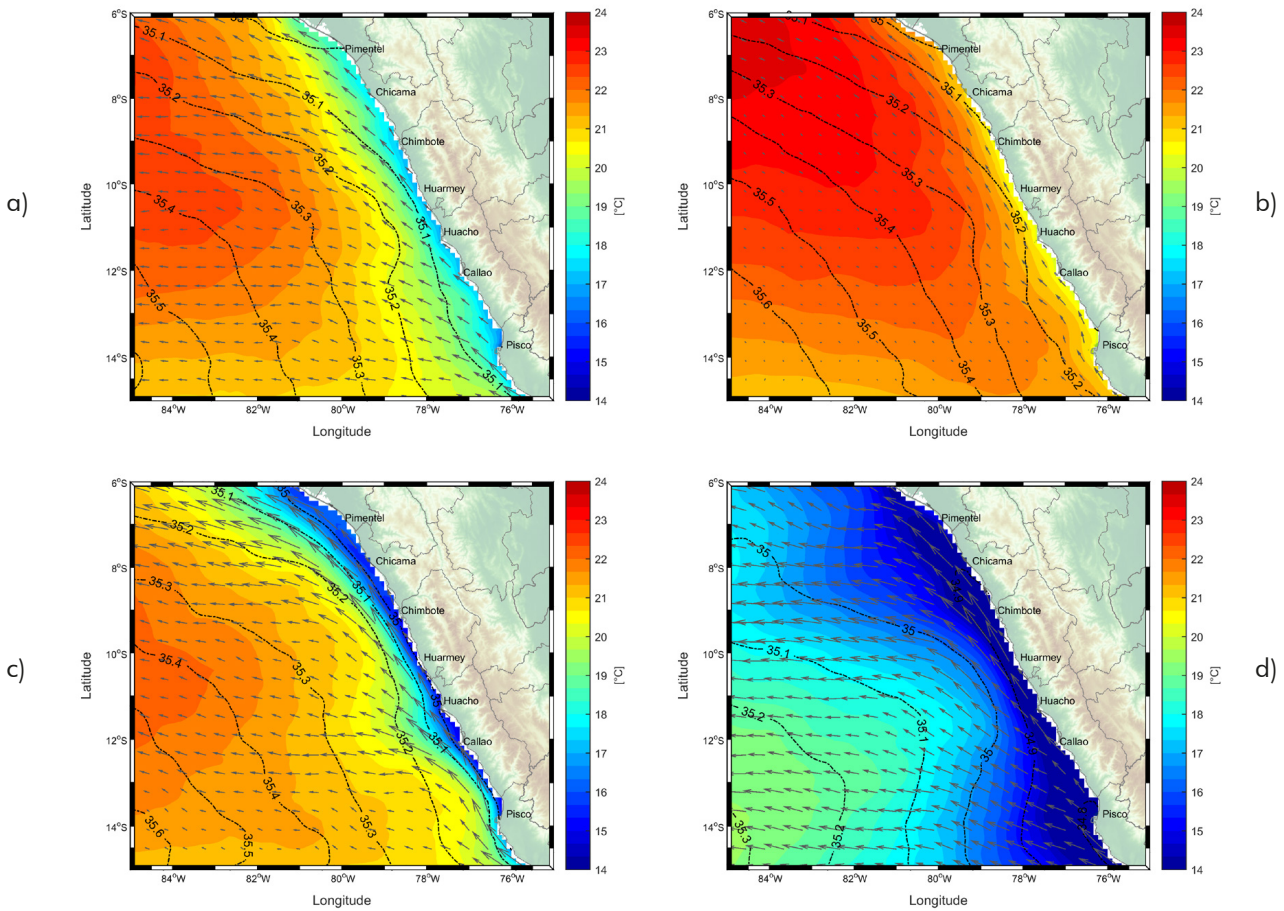


Figura SEQ Figura * ARABIC 2. Distribución horizontal del promedio anual de la temperatura superficial del mar [°C] para los escenarios de RSoda (a), RSoda_Qmin (b), RSoda_Qmax (c) y RSoda_Q5(d). Las líneas solidas muestran la distribución de la Salinidad superficial [°PSU] y las flechas indican la dirección de la corriente superficial.

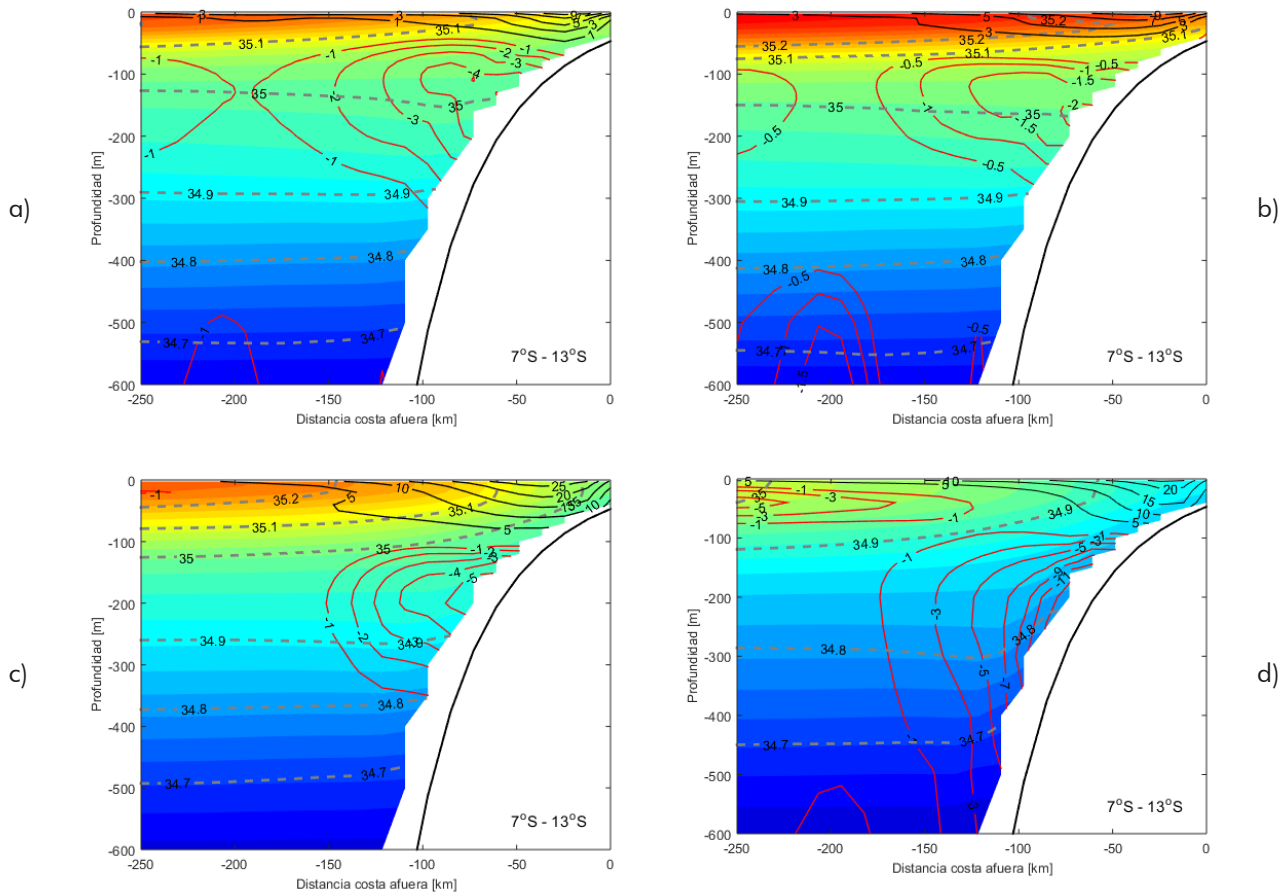


Figura 3. Sección vertical de la Temperatura [°C] promediada latitudinalmente entre 7°S - 13°S, para los escenarios RSoda (a), RSoda_Qmin (b), RSoda_Qmax (c) y RSoda_Q5(d). Las líneas grises representan las isohalinas, mientras que las líneas rojas (negras) representan las corrientes [m.s-1] en dirección Sur (Norte).

3. RESULTADOS Y DISCUSIONES

3.1. Variaciones en Temperatura, salinidad y campo de velocidades

Similar a lo indicado por Bakun (1990), los escenarios de intensificación de vientos (RSoda_Qmax y RSoda_Q5) muestran un enfriamiento de la temperatura superficial del mar (TSM; Figura 2c y 2d).

Por otro lado, el debilitamiento de vientos (RSoda_Qmin) exhibe un incremento en la TSM en concordancia a lo indicado por Goubanova et al. (2011) y Steiner and Sou (2015). Este incremento de TSM se ve influenciada por una lengua cálida que se extiende en dirección noroeste – sudeste la parte Ecuatorial (Figura 2b), y alcanza las capas subsuperficiales (~100m de profundidad) cerca de la costa (Figura 3b).

Respecto a la circulación, la corriente subsuperficial presenta una relación directa con el esfuerzo de vientos como está descrito por previos autores (e.g. Penven et al., 2005); se debilita y se extiende longitudinalmente cuando hay debilitamiento del esfuerzo de vientos, mientras que se profundiza, intensifica y se pega hacia la plataforma cuando hay incremento del esfuerzo de vientos (Figura 3). Esto último fue también encontrado por Albert et al. (2010) quien consideró un escenario con intensificación de esfuerzo de vientos sin variación espacial. Además, de las secciones verticales (Figura 3) se puede observar que por debajo de los 550m no se

observan cambios significativos de las variables de temperatura y salinidad entre las simulaciones y RSoda.

3.2. Variaciones en la actividad de mesoescala

El algoritmo detectó para cada escenario simulado, en superficie y para todo el año climatológico:

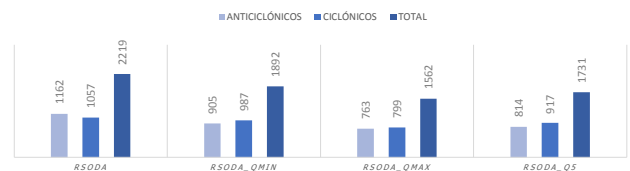


Figura 4. Gráfico de barras con el total de remolinos detectados en superficie para cada escenario de simulación.

Respecto a las características de los remolinos según Chaigneau et al. (2008) frente a la costa peruana existe mayor frecuencia de remolinos con radios (amplitudes) de ~50km (1cm), y presencia de vórtices con radios(amplitud) >175km (~5cm). Nuestro escenario control (RSoda) muestra mayor frecuencia de remolinos con radio (amplitudes) <50km (<1cm) y presencia de vórtices con radios ~160km (Figura 4 y 5). No obstante, se debe considerar que al área de estudio de Chaigneau, abarca toda la costa peruana, asimismo, para el cálculo de las amplitudes Chaigneau excluye los remolinos con radios <35km.

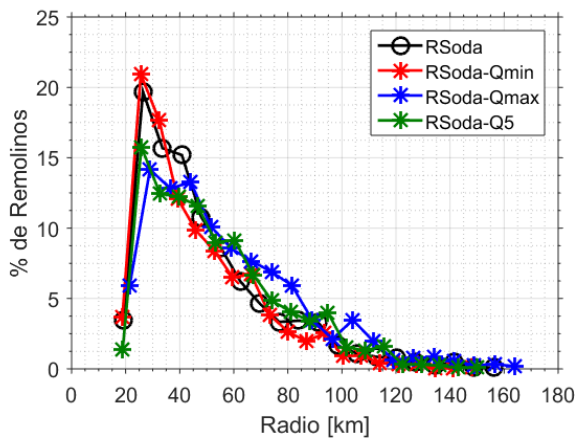


Figura 4. Distribución de radios de remolinos [km] vs. porcentaje de remolinos detectados. Los círculos negros corresponden a RSoda, y los asteriscos corresponden a los escenarios: RSoda_Qmin (rojas), RSoda_Qmax (azules), y RSoda_Q5 (verdes).

Para RSoda_Qmin hay mayor cantidad de remolinos con radios de ~ 30 km (amplitud de ~ 0.34 cm), y presencia de vórtices con radio (amplitud) de ~ 110 km (~ 5 cm). Para RSoda_Qmax hay cantidad de remolinos con radios (amplitudes) < 50 km (~ 0.67 cm), y presencia de vórtices con radios (amplitudes) de > 160 km (~ 9 cm) (Fig. 5, líneas azules). Para RSoda_Q5 hay mayor cantidad de remolinos con radios > 50 km, asimismo existe presencia de remolinos con amplitudes ~ 11 cm (Fig. 5, líneas verdes).

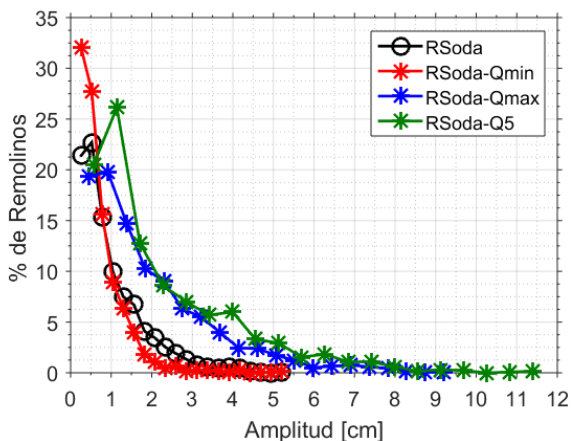


Figura 5. Distribución de amplitud de remolinos [cm] vs. porcentaje de remolinos detectados. Los círculos negros corresponden a RSoda, y los asteriscos corresponden a los escenarios: RSoda_Qmin (rojas), RSoda_Qmax (azules), y RSoda_Q5 (verdes).

Referente al índice de actividad del remolino (EAI), el algoritmo calculó para RSoda $18.8 \cdot 10^{-2} \text{cm}^2 \text{s}^{-2} \text{km}^2$, para RSoda_Qmin $10.68 \cdot 10^{-2} \text{cm}^2 \text{s}^{-2} \text{km}^2$, para RSoda_Qmax $27.46 \cdot 10^{-2} \text{cm}^2 \text{s}^{-2} \text{km}^2$ y para RSoda_Q5 $37.37 \cdot 10^{-2} \text{cm}^2 \text{s}^{-2} \text{km}^2$.

4. CONCLUSIONES Y RECOMENDACIONES

Los resultados muestran que, a pesar de las diferencias en el forzante de vientos, la acción de viento estaría influenciando la capa localizada sobre los ~ 550 m de profundidad. Asimismo, los cambios de temperatura en el volumen del dominio entre una u otra simulación son menores a $\pm 2^\circ\text{C}$ respecto al escenario de control. Con respecto a la circulación, el análisis de sensibilidad muestra que la PCUC es claramente afectada en su extensión e intensidad por la acción del viento, siendo más intensa y plegada a la plataforma cuando mayor es el esfuerzo del

viento. Respecto a la actividad de mesoescala la variación de los vientos (intensificación o debilitamiento) exhibe menor cantidad de remolinos en total, sin embargo los cambios drásticos son respecto a sus características (Radio, Amplitud, Densidad de Energía (EI) e Índice de Actividad de remolino (EAI)); presentando mayor frecuencia de remolinos con radios y EAI mayores cuando mayor es el esfuerzo de viento, debido a la variación de energía cinética asociada.

Si bien esta tesis se genera a partir de hipótesis de variación de vientos, no se contempla variación en el flujo de calor y aguas frescas. Por tanto, se sugiere tener cuidado con las diferentes interpretaciones derivadas de este trabajo.

5. AGRADECIMIENTOS

Al Instituto Geofísico del Perú y a la Dra. Ivonne Montes por brindarme las facilidades para desarrollar mi tema de tesis.

Este trabajo utilizó los recursos computacionales, HPC-Linux-Cluster, del Laboratorio de Dinámica de Fluidos Geofísicos Computacionales del Instituto Geofísico del Perú (Subvenciones 101-2014-FONDECYT, SPIRALES2012 IRD-IGP, Manglares IGP-IDRC, PpR068).

6. BIBLIOGRAFÍA

- Albert, A., Echevin, V., Lévy, M., and Aumont, O. (2010). Impact of nearshore wind stress curl on coastal circulation and primary productivity in the Peru upwelling system. *Journal of Geophysical Research: Oceans*, 115(12), 1–13. <https://doi.org/10.1029/2010JC006569>
- Bakun, A. (1990). Global climate change and intensification of coastal ocean upwelling. *Science*, 247(4939), 198–201. <https://doi.org/10.1126/science.247.4939.198>
- Chaigneau, A., Gizolme, A., and Grados, C. (2008). Mesoscale eddies off Peru in altimeter records: Identification algorithms and eddy spatio-temporal patterns. *Progress in Oceanography*, 79(2–4), 106–119. <https://doi.org/10.1016/j.pocean.2008.10.013>
- Colas, F., Capet, X., McWilliams, J. C., and Li, Z. (2013). Mesoscale Eddy Buoyancy Flux and Eddy-Induced Circulation in Eastern Boundary Currents. *Journal of Physical Oceanography*, 43(6), 1073–1095. <https://doi.org/10.1175/JPO-D-11-0241.1>
- Goubanova, K., Echevin, V., Dewitte, B., Codron, F., Takahashi, K., Terray, P., and Vrac, M. (2010). Statistical downscaling of sea-surface wind over the Peru-Chile upwelling region: Diagnosing the impact of climate change from the IPSL-CM4 model. *Climate Dynamics*, 36(7), 1365–1378. <https://doi.org/10.1007/s00382-010-0824-0>
- Montes, I., Colas, F., Capet, X., and Schneider, W. (2010). On the pathways of the equatorial subsurface currents in the eastern equatorial Pacific and their contributions to the Peru-Chile Undercurrent. *Journal of Geophysical Research: Oceans*, 115(9), 1–16. <https://doi.org/10.1029/2009JC005710>
- Shchepetkin, A.F., and McWilliams, J.C. (2003). A method for computing horizontal pressure-gradient force in an oceanic model with a nonaligned vertical coordinate. *J. Geophys. Res.*, 108(C3), 3090, <https://doi.org/10.1029/2001JC001047>
- Xu, C., Zhai, X., and Shang, X. (2016). Work done by atmospheric winds on mesoscale. *Journal of Geophysical Research: Oceans*, 1–7. <https://doi.org/10.1002/2016GL071275>
- Yi, J., Du, Y., He, Z., and Zhou, C. (2014). Enhancing the accuracy of automatic eddy detection and the capability of recognizing the multi-core structures from maps of sea level anomaly. *Ocean Science*, 10, 39–48. <https://doi.org/10.5194/os-10-39-2014>

CARACTERIZACIÓN DEL ROL DEL PÁRAMO ANDINO EN LA REGULACIÓN DEL RECURSO HÍDRICO EN LA CUENCA DEL RÍO RONQUILLO

**Characterization the role of the Andean Paramo on the water resources regulation in the Ronquillo river basin*

Jhon Dany Orrillo Vigo

Universidad Nacional de Cajamarca
Facultad de Ingeniería – Escuela Académico
Profesional de Ingeniería Hidráulica
jorrillo@igp.gob.pe

Asesor: Dr. Sergio Byron Morera Julca

Subdirección de Ciencias de la Atmósfera e
Hidrosfera - IGP

RESUMEN

La cuenca Ronquillo abastece con un 30% al sistema de agua potable de la ciudad de Cajamarca. En esta región no existen glaciares, se presume que los ecosistemas de montaña, e.j. los Páramos Andinos proveen de recurso hídrico durante la época de estiaje. Aplicando el Índice de Flujo Base (BFI) en la cuenca del río Ronquillo, así como las subcuencas Sexemayo y Majadapampa, ambas con predominancia de suelos páramos de 4.7 km² y 6.8km², respectivamente; se identificó: (i) la cuenca Sexemayo presenta los valores más altos de BFI=0.72, (ii) la cuenca Majadapampa muestra un BFI igual a 0.68, y (iii) la cuenca Ronquillo muestra los valores más bajos de BFI=0.5. A partir de los resultados concluimos que los suelos páramo regulan hasta el 72% del paso del recurso hídrico en una cuenca de montaña.

Palabras clave: Páramo Andino, Caudal base, Índice de Flujo Base, Filtros digitales.

ABSTRACT

The Ronquillo watershed supplies 30% of the drinking water in the Cajamarca. In this region, there are not glaciers, and mountain ecosystems, e.g. Andean paramo, are expected to supply water during the dry season. Applying the Base Flow Index (BFI) at the Ronquillo watershed, as well as, the Sexemayo and Majadapampa sub-watershed, both with predominance of Páramos soil of 4.7 km² and 6.8km²; we reveal: (i) Sexemayo sub-watershed shows the highest base flow rate equal to 0.72, (ii) Majadapampa sub-watershed shows the BFI equal to 0.68 and (iii) the Ronquillo watershed show the lowest value (BFI=0.50). From these result we conclude that Páramos soil (moorland) regulate up to 72% of the water resources within a mountain watershed.

Keywords: Páramo Andino, Base flow, Base flow index, Digital filters.

1. INTRODUCCIÓN

Los Páramos son ecosistemas con un alto potencial para la regulación del recurso hídrico, de esta manera sus suelos logran retener agua en época de lluvias, para luego dejarlas fluir en meses sin precipitación. Por otro lado, los suelos característicos de los Páramos, son altamente vulnerables a las actividades antrópicas, las cuales las pueden afectar de manera irreversible.

La cuenca del Ronquillo (42 km²) es de gran relevancia para esta región (Figura 1), ya que proporciona el 30% del agua potable

a la ciudad de Cajamarca. En la época de estiaje experimenta una marcada escasez del recurso hídrico que se agudiza en época de estiaje, esta problemática afecta a la población que habita dentro de la cuenca, así como a la población de la ciudad de Cajamarca. Por tal razón, el objetivo de la presente investigación fue cuantificar la influencia del páramo andino en la regulación del recurso hídrico en la cuenca Ronquillo, colaborando a su puesta en valor.

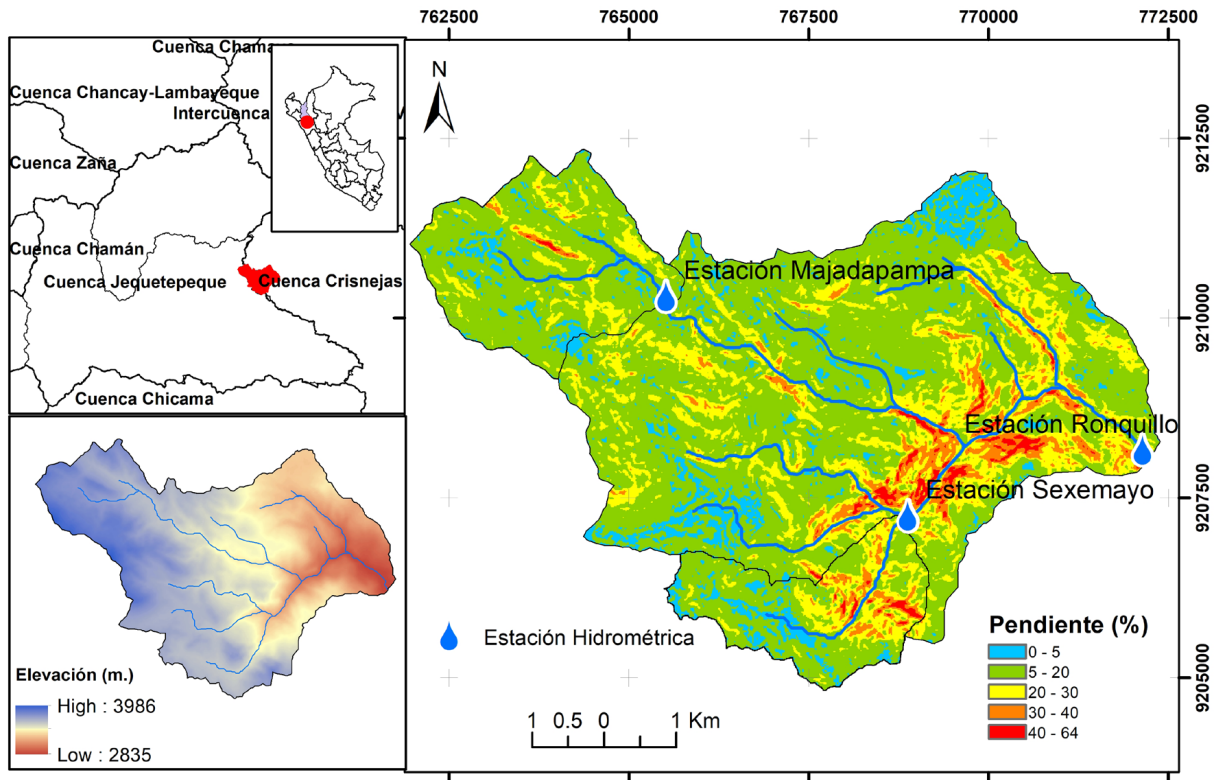


Figura 1. Ubicación de la cuenca Ronquillo.

2. DESARROLLO

2.1. Monitoreo del sedimento de fondo.

El desarrollo de la investigación se resume en la Figura 2, iniciando con la (1) recolección de los datos, pasando luego a una etapa de (2) procesamiento que involucra separar el caudal base con las diferentes metodologías existentes, luego se para a una esta importante como es la (3) calibración para intentar que nuestras metodologías se asemejen a la realidad, y finalmente (4) análisis de los resultados.

Se utilizaron caudales estimados obtenidos a partir de los niveles registrados cada 5 minutos, en las estaciones de Majadapampa, Sexemayo entre los años 2016 y 2017, y entre 2008 y 2017 para la cuenca Ronquillo. Se realizó la separación de caudal base del caudal total, utilizando ocho métodos de separación. Estos métodos están agrupados en métodos gráficos o también llamados de filtrado y los métodos de filtros digitales. Luego de realizar la separación del caudal base, se escogió los métodos

más consistentes para realizar su calibración, estos métodos seleccionados fueron los métodos de filtros digitales, para la calibración se utilizó desviación media cuadrática para toda la serie de datos (RMSD) y solo para el periodo de estiaje (RMSD Q). Esta información sirvió para calcular el índice de flujo base (BFI), el cual es la relación entre el volumen de caudal base y el volumen de caudal total, para la cuenca Ronquillo, subcuencas de Sexemayo y Majadapampa.

Dentro de los dos grupos de métodos utilizados, para la separación del caudal base, se tiene: (i) los métodos de filtros digitales, donde se utilizaron el método de Eckhardt (Eckhardt, 2008), métodos de Un parámetro (Chapman, 1999) y el método de Lyne y Hollick (Lyne & Hollick, 1979), de estos el método de Eckhardt se adapta mejor al área en estudio. Por su parte (ii) los métodos gráficos (Sloto & Crouse, 1996) que se basan en identificar valores mínimos del hidrograma en un intervalo dado de tiempo (N), sobrestimando el caudal base para nuestra zona de estudio.

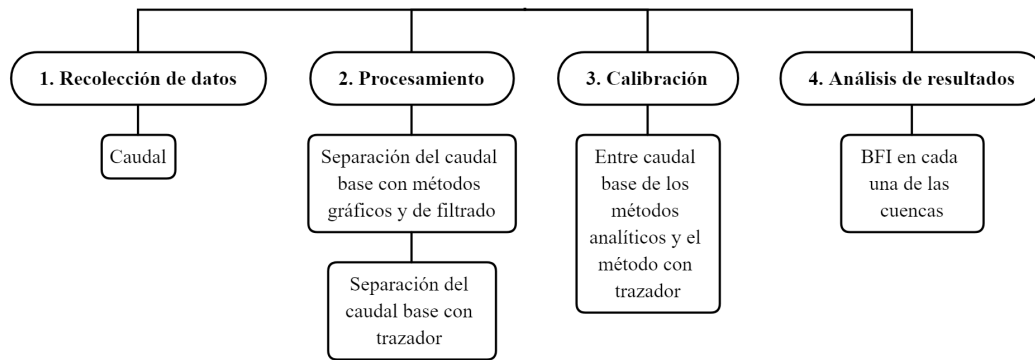


Figura 2. Esquema de la metodología

3. RESULTADOS.

3.1. Separación del caudal base e Índice de Flujo Base (BFI)

3.1.1. Cuenca Ronquillo

Luego de separar el caudal base y calibrar los métodos de filtros digitales, se obtuvieron los siguientes valores de BFI.

Método	BFI	RMSD	RMSD Q
Lune y Hollick	0.19	0.013	0.032
Un Parámetro	0.29	0.009	0.036
Eckhardt	0.50	0.009	0.034

Tabla 1. Resultados de valores de índice de flujo base para la cuenca Ronquillo.

El método con mejor ajuste, fue el método de Eckhard, con este método se obtuvo un BFI de 0.5 (50%).

3.1.2. Subcuenca Sexemayo

El valor de BFI obtenido para la subcuenca Sexemayo fue de 0.72 (72%), el cual significa que solo un 28% del agua precipitada llega en forma de escorrentía superficial al curso principal de agua, la parte restante se infiltra.

Método	BFI	RMSD	RMSD Q
Lune y Hollick	0.69	0.229	0.007
Un Parámetro	0.50	0.140	0.023
Eckhardt	0.72	0.188	0.012

Tabla 2. Resultados de valores de índice de flujo base para la subcuenca Sexemayo

3.1.3. Subcuenca Sexemayo

Para la estación Majadapampa (Tabla 3), el filtro de de Lyne y Hollick calcula un índice de flujo base igual a 0.88, con el RMSD más bajo. Este valor de BFI es un valor muy alto, considerando que es un río con un régimen efímero.

Aplicando el filtro de Eckhardt, el valor de índice de flujo base es igual a 0.68, con un valor de RMSD igual a 0.055 m³/s, que es un valor intermedio considerando los dos filtros antes detallados.

Método	BFI	RMSD
Lune y Hollick	0.88	0.041
Un Parámetro	0.50	0.059
Eckhardt	0.68	0.055

Tabla 3. Resultados de valores de índice de flujo base para la subcuenca Majadapampa.

4. DISCUSIÓN

4.1. Volumen de caudal base y escorrentía superficial

Luego de realizar la separación del caudal base, se calculó el volumen de la escorrentía superficial y el caudal base. En la cuenca Ronquillo, para los años hidrológicos entre 2009 al 2016 se puede ver la variación de volumen de agua (escorrentía superficial y caudal base), de esto se infiere que el año hidrológico 2016 fue el año con menor volumen de agua producido por la cuenca Ronquillo con 6.6 millones de metros cúbicos, mientras que el año hidrológico 2015 con 15.3 millones de metros cúbicos fue el año con mayor producción de volumen de agua. También se puede ver (Figura 3) que luego de un año con alta producción de agua, se presenta años con una producción menor de volumen de agua.

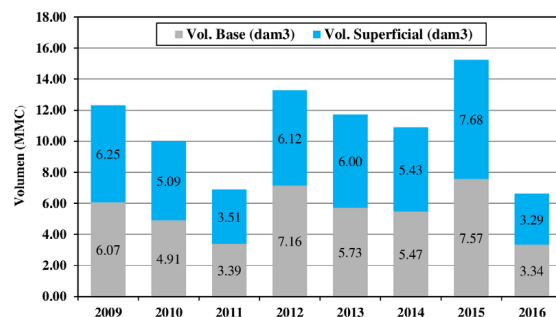


Figura 3. Volumen de caudal base (color gris) y caudal de escorrentía superficial (color celeste), para los años hidrológicos desde el 2009 al 2016.

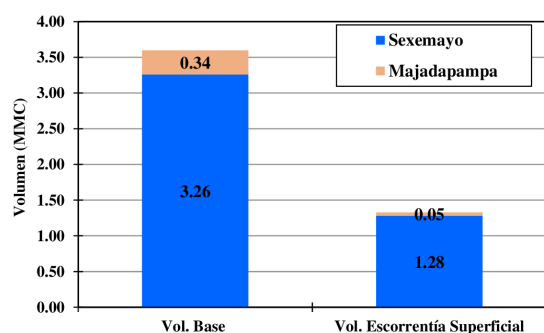


Figura 4. Volúmenes de agua de caudal base para las subcuencas de Sexemayo (color azul) y Majadapampa (color naranja).

Para las subcuencas Sexemayo y Majadapampa se observa (Figura 4) que los volúmenes de agua en ambas subcuencas difieren de forma considerable, tanto en volumen base (a partir del caudal base) como el volumen de escorrentía superficial (caudal directo). Pero tanto en la subcuenca Sexemayo como Majadapampa el volumen base es mayor que el volumen de la escorrentía superficial.

Según los datos registrados, en cada punto emisor de las subcuencas sexemayo y Majadapampa, los volúmenes son 4.5 millones m³ y 3.9 millones m³ respectivamente, a pesar de que el área de la subcuenca Sexemayo (4.7 km²) es más pequeña que el de la subcuenca Majadapampa (6.8 km²), ambas presentan diferencias considerables en cuanto a los volúmenes de agua producidos en cada subcuenca, que no son una relación directamente proporcional al área. Esta diferencia podría ser causada por la presencia de roca altamente fracturada en la parte de Majadapampa y por la presencia de una falla geológica paralela al curso de agua.

5. CONCLUSIONES

La cuenca Ronquillo presenta un BFI de 0.50, esto es producto de su variada geología y tipos de suelos que van desde Andosol, con características buenas para la retención de agua, hasta suelos Leptosol con escasa capacidad de retención de humedad. Las subcuencas de Sexemayo y Majadapampa, ambas con predominancia de depósitos de suelo Andosol, Páramo Andosol y Paramosol, presentan un BFI igual a 0.72 y 0.68, respectivamente. En otras palabras, su capacidad de regulación del recurso hídrico es del 72% y 68% de la precipitación acumulada en un año en la cuenca.

Se monitoreó la humedad de un suelo tipo Andosol/Páramo, se encontró que las características especiales (estructura del suelo y vegetación) permiten que solo pierda humedad los 25 cm superficiales en agosto-septiembre (periodo de estiaje), y bajo los 75 centímetros se mantiene con alta humedad durante todo el año. Al relacionar el BFI con la humedad se encontró que a mayor contenido de humedad se tiene mayor valor de BFI.

6. RECOMENDACIONES

Los suelos de tipo Andosol son buenos almacenadores de agua, pero muy sensibles al cambio de uso de suelo, en tal sentido es importante realizar más investigaciones para la puesta en valor de los Páramos de la cuenca Ronquillo, para poder reducir los cambios de usos de suelo, que en la actualidad ya se vienen dando, y con esto evitar que estos suelos se afecten de forma irreversible.

Cuando se pretende realizar una separación de caudal base, se debe trabajar con varios métodos de separación, para poder analizar cuál se adapta mejor al área de estudio, incluyendo siempre métodos gráficos, métodos de filtros digitales y algún método geoquímico.

7. AGRADECIMIENTOS

Este estudio se realizó en el Instituto Geofísico del Perú, y fue apoyado por el programa presupuestal PPR-068 "Reducción de Vulnerabilidad y Atención de Emergencias por Desastres" y Fondecyt del CONCYTEC a través del proyecto "Influencia de los páramos en la erosión y conservación de suelos andinos, e identificación de su rol en la regulación del recurso hídrico en grandes ciudades altoandinas" (Contrato N° 098-2015-FONDECYT). Además, agradezco a la E.P.S. SEDACAJ, por el apoyo brindado durante la campaña de campo.

8. BIBLIOGRAFÍA

- Chapman, T. (1999). A comparison of algorithms for streamflow recession and baseflow separation. *Hydrological Processes*, 13(July 1998), 701-714. [https://doi.org/10.1002/\(SICI\)1099-1085\(19990415\)13:5<701::AID-HYP774>3.0.CO;2-2](https://doi.org/10.1002/(SICI)1099-1085(19990415)13:5<701::AID-HYP774>3.0.CO;2-2)
- Eckhardt, K. (2008). A comparison of baseflow indices, which were calculated with seven different baseflow separation methods. *Journal of Hydrology*, 352(1-2), 168-173. <https://doi.org/10.1016/j.jhydrol.2008.01.005>
- Lyne, V. D. D., & Hollick, M. (1979). Stochastic time-variable rainfall-runoff modelling. *Hydrology and Water Resources Symposium*.
- Sloto, R. a., & Crouse, M. Y. (1996). Hysep: a computer program for streamflow hydrograph separation and analysis. U.S. Geological Survey Water-Resources Investigations Report 96-4040, 54.

ESTUDIO DE TRANSPORTE DE SEDIMENTOS EN LA CUENCA DEL RÍO RONQUILLO

**Sediment transport study in the Ronquillo river basin*

Diana Violeta Díaz Llatas

Universidad Nacional de Cajamarca
Facultad de Ingeniería – Escuela Académico
Profesional de Ingeniería Hidráulica
dianadiaz@outlook.es

Asesor: Dr. Sergio Byron Morera Julca

Subdirección de Ciencias de la Atmósfera e
Hidrosfera - IGP

RESUMEN

En la presente investigación se realizaron mediciones de sedimento de fondo mediante una estrategia de monitoreo de sedimento de fondo, utilizando la presa de derivación Ronquillo como un acumulador o totalizador de sedimentos de capacidad de almacenamiento conocido, se registraron los periodos de llenado del totalizador, caudales y tiempo involucrados. Con esta estrategia se registraron los volúmenes de sedimentos de fondo medidos durante el periodo 2008 - 2016. Se seleccionaron 4 fórmulas empíricas: Parker, Klingteman y McLean 1982 (P-K-Mc), Smart y Jaeggi 1983 (S-J), Bathurst (1987) y Meyer Peter y Muller 1948 (M-P-M). Los resultados obtenidos con cada una de ellas, se compararon con el volumen de sedimento observado en campo, todas las fórmulas utilizadas sobre estimaron el volumen de sedimento y caudal sólido de fondo real, la fórmula de Parker, Klingteman and McLean 1982 (P-K-Mc), fue la que sobreestimó en menor cantidad al volumen real.

Palabras clave: Sedimento de fondo, sedimento en suspensión, río Ronquillo, volumen de sedimento de fondo, coeficiente de proporcionalidad, caudal sólido.

ABSTRACT

In the present research, measurements of bedload sediment were performed, through a bedload sediment monitoring strategy, using the Ronquillo bypass dam as an accumulator or totalizer of sediments of known storage capacity, the accumulator filling periods, flows and time involved were recorded. With this strategy, the bedload sediment volumes measured during the 2008 - 2016 period were recorded. Four empirical formulas were selected: Parker, Klingteman and McLean 1982 (P-K-Mc), Smart and Jaeggi 1983 (S-J), Bathurst (1987), and Meyer Peter and Muller 1948 (M-P-M). The results obtained with each of them were compared with the sediment volume observed in the field, all the formulas used on estimated the sediment and real solid flow, the Parker, Klingteman and McLean 1982 (P-K-Mc) formula, was the one he estimated to be less in real volume.

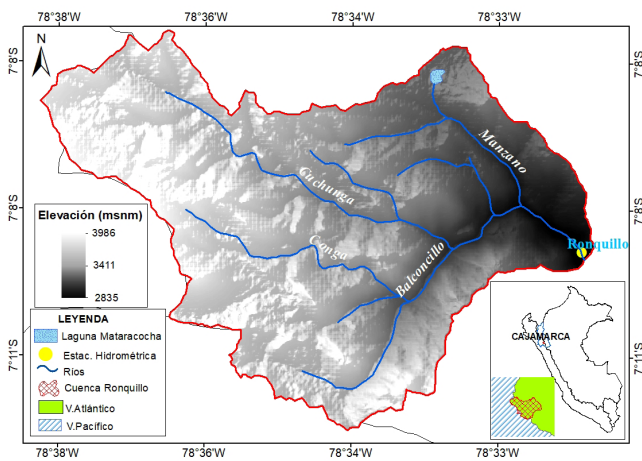
Keywords: Bedload Sediment, suspended sediment, Ronquillo river, bedload sediment volume, proportionality coefficient, solid flow.

1. INTRODUCCIÓN

La presente investigación estableció una técnica de monitoreo de sedimentos de fondo para estimar el real transporte de sedimentos, para la cual se utilizó como trampa de sedimentos a la presa de derivación Ronquillo, ubicada en la estación de monitoreo. Los resultados medidos en campo, se compararon con el estimado a partir del uso de cuatro fórmulas empíricas: (i) Smart (1984); (ii) Bathurst (et. al., 1987); (iii) Sub-sustrato basado en la ecuación de Parker, Klingeman, & McLean, (1982), y (iv) Meyer-Peter & Müller, (1948).

La investigación se realizó en la Cuenca del río Ronquillo (41.7 km², 3508 m s.n.m. y 7.1 °S y 78.5 °W), ubicada en el norte de Perú, en el distrito de Cajamarca, al este de la ciudad de Cajamarca (Figura 1).

El tramo del río Ronquillo estudiado corresponde a 200 m aguas arriba de la estación hidro-sedimentológica Ronquillo, se seleccionó este tramo por estar ubicado cerca al punto emisor de la Cuenca, ser representativo para realizar el muestreo de granulometría y obtener los parámetros necesarios del cauce (Tabla 1).



Los diámetros característicos del material del lecho sub superficial del río Ronquillo son: D16=2.12 mm, D50=8.77 mm, D84=31.64 mm, D90= 39.82 mm y Dm=16.26 mm, mientras que los diámetros característicos del lecho superficial son: D16=29.59 mm, D50=94.73 mm, D84=343.84 mm, D90= 446.22 mm y Dm= 129.48 mm.

	Parámetros	Valor
Cuenca Ronquillo	Área de la cuenca, Km ²	41.7
	Perímetro de la cuenca, Km	34.5
	Pendiente promedio de la cuenca, %	0.29
	Elevación máxima, m s.n.m.	3985
	Elevación media, m s.n.m.	3508
	Elevación mínima, m s.n.m.	2835
	Tiempo de concentración, mín.	235
	Longitud del máximo recorrido, Km	13.2

Río Ronquillo	Pendiente media del río ubicado aguas arriba de la captación, %	3.16
	Precipitación anual, mm	1280
	Descarga máxima medida, m ³ s ⁻¹	33.7
	Descarga media medida, m ³ s ⁻¹	0.2
	Descarga mínima medida, m ³ s ⁻¹	0.1

Tabla 1. Principales características de la cuenca y río Ronquillo

2. DESARROLLO

2.1. Monitoreo del sedimento de fondo

El punto de monitoreo está ubicado en la presa de derivación Ronquillo (ubicada en el punto emisor de la Cuenca del mismo nombre).

La estructura permite la formación de una presa con una distancia de 16 m aguas arriba, su capacidad máxima de almacenamiento corresponde a 68±2 m³, esta presa actúa como un totalizador de sedimentos ya que en ella se acumulan los sedimentos transportados y cuando estos llegan al nivel máximo de acumulación, se abren las compuertas de limpieza (CG-1 y CG-2, ver Figura 2) para eliminar los sedimentos de la presa y dejarla limpia.

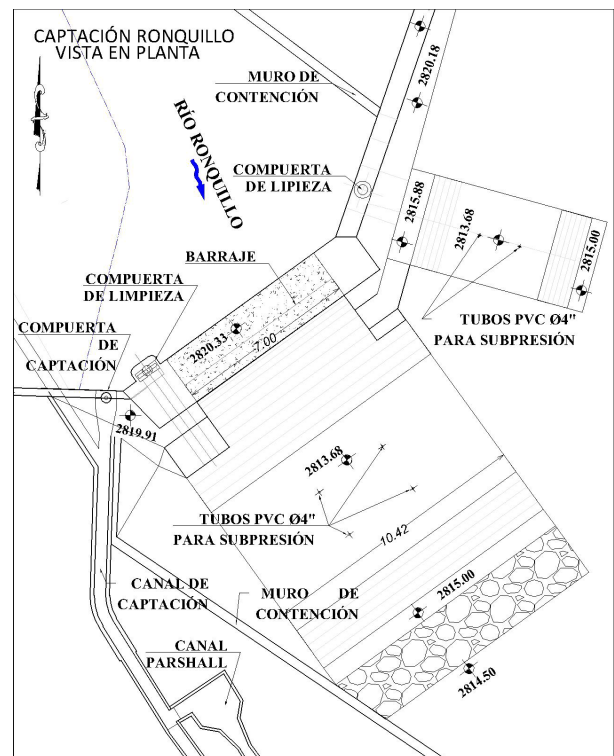


Figura 2. Estructura de captación Ronquillo, se muestra el canal de captación y el barraje de la presa en la cual se acumulan los sedimentos de fondo. Las compuertas CG-1 y CG-2 son utilizadas para realizar la limpieza del vaso de la presa cuando este se llena de sedimento de fondo.

Fuente: Modificado de SEDACAJ

Este tipo de monitoreo es un método directo semejante a las trampas de sedimentos, las que consisten en una trampa ubicada en el cauce del río, en la cual se acumulan los sedimentos (Hudson, 1997).

2.2. Estimación de transporte de sedimentos utilizando fórmulas empíricas

Para la estimación del sedimento de fondo, se utilizaron 4 fórmulas empíricas (Tabla 2): (i) Fórmula del Sub-sustrato basado en la ecuación de Parker, Klingeman y McLean, (1982) (P-K-Mc), que calcula las tasas de transporte sobre la base de un único

tamaño de grano (D50 subsuperficial); (ii) Smart, (1984) (S-J), esta fórmula se realizó con 213 pruebas con pendientes entre 0.04% - 20%; (iii) Bathurst, (1987), que considera el exceso de caudal sobre el caudal crítico requerido para iniciar el movimiento de las partículas y (iv) Meyer -Peter Y Muller, (1948) (M-P-M), realizada en el laboratorio de investigación hidráulica de Zurich, con material no uniforme de varias densidades.

Autor	Fórmula	Observación
Sub-sustrato basado en la ecuación de Parker, Klingeman, & McLean, (1982) (P-K-Mc)	$W^* = \left\{ \begin{array}{l} ; \\ ; \\ ; \\ \tau_{50}^* = \frac{u_*^2}{(s-1)gD_{50sub}} \end{array} \right. ; \quad \phi_{50} = \frac{\tau_{50}^*}{\tau_{r50}^*}$ $Q_b = \frac{W^* u_*^3 \rho_s}{(s-1)g}$	Desarrollado para ríos con lecho de grava.
Smart, (1984) (S-J)	$\Phi^* = 4 \left[\frac{D_{90}}{D_{30}} \right]^{0.2} S^{0.6} C^* \tau^* \left[\tau^* - \tau_{cj}^* \right]$ $\tau_{cj}^* = \tau_c^* \cos(\arctan S) \left[1 - \frac{S}{\tan \phi} \right]$ $\Phi^* = \frac{q_s}{D \sqrt{g \Delta D}}$	0.4% ≤ S ≤ 20% Diámetros > 0.4 mm
Bathurst et al., (1987)	$\Phi^* = \frac{2.5 S^{3/2}}{(\Delta+1)D_{50} \sqrt{g \Delta D_{50}}} (q - q_c)$ $\frac{q_c}{\sqrt{g D_{50}^3}} = 0.15 S^{-1.12}$ $\frac{q_c}{\sqrt{g D_{16s}^3}} = 0.21 S^{-1.12}$ $\Phi^* = \frac{q_s}{D \sqrt{g \Delta D}}$	0 0.4% ≤ S ≤ 4% 0.30 ≤ d/D ₈₄ ≤ 50
Meyer-Peter & Müller, (1948) (M-P-M)	$Q_b = 8 \gamma_s \left(g \left(\frac{\gamma_s}{\gamma_s - \gamma} D_m^3 \right) \right)^{0.5} \left[\left(\frac{n_s}{n} \right)^{1.5} \tau_* - 0.047 \right]^{1.5}$ $n_s = \frac{D_{90s}^{1/6}}{26}$	1.4 < D50 < 29mm

Tabla 2. Fórmulas para sedimento de fondo utilizadas.

Dónde:

ϕ_{50} : Parámetro adimensional en base a tensión de corte necesario para movilizar partículas de tamaño D50.

τ_{50}^* : Tensión de referencia de Shield igual a 0.0876.

τ_c^* : Esfuerzo cortante crítico.

S: Pendiente del cauce (%).

q: Caudal de agua por unidad de ancho.

q_c: Caudal líquido crítico.

q_s: Caudal sólido de fondo (m³/s-m).

Q_b: Caudal sólido de fondo (Kg/s-m).

ϕ : Ángulo de fricción interna de las partículas.

γ_s : Peso específico del suelo (Kg/m³).

γ : Peso específico del agua (Kg/m³).

D_m: Diámetro medio (m).

D_{90s}: Diámetro de la partícula del 90% de sedimento superficial.

D₅₀: Diámetro del 50% del material del lecho sub-superficial.

D₃₀: Diámetro del 30% del material del lecho sub-superficial.

D_{16s}: Diámetro del 16% del material del lecho superficial.

D : Diámetro característico del lecho del cauce.

τ_* : Número adimensional de Shields.

- n_s : Coeficiente de manning debido a la partícula.
- n : Coeficiente de manning.
- g : Aceleración debido a la gravedad (m2/s).
- s : Relación entre peso específico del suelo y del agua γ_s/γ .
- Δ : Corresponde a $s-1$.

2.3. Curva de gasto con valores estimados y observados

Esta curva consiste en la relación entre la descarga de agua y la descarga de sedimentos (Glysson & Survey, 1987).

$$Q_s = aQ^b \quad (1)$$

Donde:

Q_s = Descarga de sedimento o caudal sólido (t/día)

Q = Caudal de descarga ($m^3 s^{-1}$)

a y b = Coeficientes de ajuste de la correlación.

3. RESULTADOS

3.1. Comparación de volúmenes de sedimentos observados y estimados

Para el análisis de los datos obtenidos, se separaron en 2 grupos, teniendo en cuenta la duración de cada evento de crecida que llenó con sedimentos la presa de derivación Ronquillo. El "Grupo 1" (Figura 3) está conformado por eventos que duraron como máximo 90 minutos, y presentan una variación de caudales líquidos desde $3 m^3 s^{-1}$ hasta los $20 m^3 s^{-1}$, mientras que el "Grupo 2" (Figura 4) está conformado por los eventos de crecida que duraron más de 90 minutos, conformado por caudales que varían desde $3 m^3 s^{-1}$ hasta los $34 m^3 s^{-1}$.

El coeficiente de proporcionalidad (Ve/Vm), corresponde al volumen de sedimento de fondo estimado con las fórmulas empíricas (Ve) entre el volumen de sedimento de fondo medido en campo (Vm).

Los coeficientes de proporcionalidad medios obtenidos para cada fórmula aplicada en ambos grupos están representados por la línea negra y valor en cada diagrama de cajas.

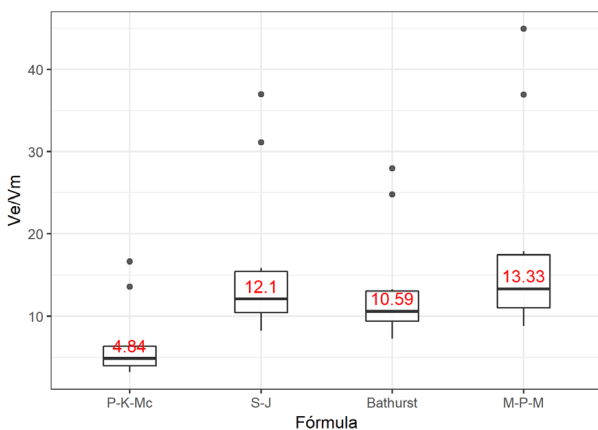


Figura 3. Coeficiente de proporcionalidad entre el volumen estimado (Ve) y el volumen medido en campo (Vm) correspondientes a los eventos que conforman el grupo 1.

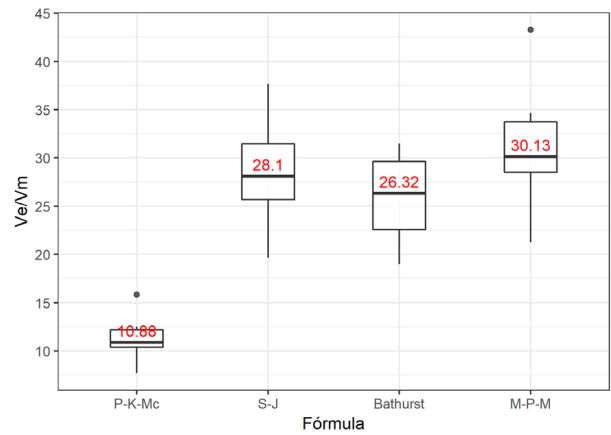


Figura 4. Coeficiente de proporcionalidad entre el volumen estimado (Ve) y el volumen medido en campo (Vm) correspondientes a los eventos que conforman el grupo 2.

3.2. Comparación de caudales sólidos observados y estimados

Al realizar el ajuste a una curva potencial de los caudales sólidos estimados con las fórmulas empíricas y observados en campo, se obtuvo los resultados mostrados en la Figura 5.

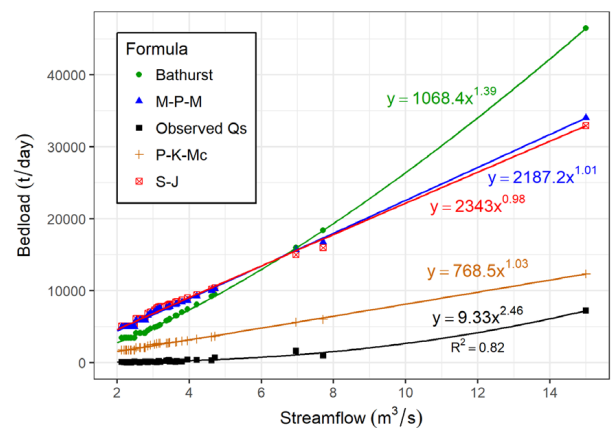


Figura 5. Curva de tendencias del sedimento de fondo observado y las tendencias del sedimento de fondo estimadas utilizando las fórmulas empíricas mostradas en la tabla 2.

4. DISCUSIÓN

Los resultados obtenidos al comparar los volúmenes de sedimentos estimados y observados (Figura 3 y Figura 4), muestran que para las 4 fórmulas los coeficientes obtenidos son mayores a 1, lo cual nos indica que los valores estimados por cada una de las fórmulas están sobre estimando los volúmenes de sedimento acumulados observados, lo mismo sucede al momento de comparar los caudales sólidos.

En la Figura 5, el coeficiente "a" representa la disponibilidad de sedimentos en la cuenca, observando que en el caso de las fórmulas empíricas ($a = 768.5$ hasta 2343) son mucho mayores que el observado ($a = 9.3$), en las fórmulas empíricas consideran mayor disponibilidad de sedimento que el real.

El exponente "b", representa la intensidad de las fuerzas erosivas, que según Iroumé (1990), varía desde 2 hasta 2.5, siendo el último valor el valor que representa la intensidad más alta. En las fórmulas empíricas el valor de "b" varía entre 0.98 y 1.39, mientras que del observado cercano es de 2.46. Este resultado claramente muestra que el río andino de montaña muestra poderosas condiciones erosivas.

5. CONCLUSIONES

Al comparar los volúmenes de sedimentos de fondo y caudales sólidos monitoreados en campo con los estimados, las fórmulas empíricas sobre-estimaron los volúmenes monitoreados. La fórmula empírica cuyos resultados se acercaron más a los reales, fue la de Parker, Klingteman y McLean, fórmulas diseñadas para cauces con lechos gravosos.

Los parámetros "a" y "b" del ajuste potencial de caudales sólidos de fondo medido en campo, mostraron que el río Ronquillo tiene una alta intensidad de fuerzas erosivas, las cuales pueden estar limitadas por la disponibilidad de sedimentos.

6. RECOMENDACIONES

Los resultados obtenidos son válidos para los caudales del periodo 2008-2016, ya que, con caudales mayores a los monitoreados en este periodo, el resultado puede variar.

Es de gran importancia continuar con el monitoreo de los sedimentos en cuencas de montaña pues estas proveen de grandes volúmenes de agua a la costa peruana y a la llanura amazónica. El obtener datos reales permitirá entender la dinámica de la cuenca en estudio y por lo tanto tomar decisiones acertadas en el futuro.

El presente estudio demuestra que las fórmulas empíricas no siempre permiten obtener resultados acordes a la realidad de la zona de estudio y realza la importancia del monitoreo de sedimentos en las cuencas.

7. AGRADECIMIENTOS

Este estudio se realizó en el Instituto Geofísico del Perú y fue apoyado por el programa presupuestal PPR-068 "Reducción de Vulnerabilidad y Atención de Emergencias por Desastres" y Fondecyt del CONCYTEC a través del proyecto "Influencia de los páramos en la erosión y conservación de suelos andinos, e identificación de su rol en la regulación del recurso hídrico en grandes ciudades altoandinas" (Contrato N° 098-2015-FONDECYT). Además, agradezco a la E.P.S. SEDACAJ, por el apoyo brindado.

8. BIBLIOGRAFÍA

Bathurst, J. C., Graf, W. H., & Cao, H. H. (1987). Bed load discharge equations for steep mountain rivers. *Sediment Transport in Gravel-Bed Rivers*. John Wiley and Sons New York. 1987. p 453-491.

Glysson, G. D., & Survey, U. S. G. (1987). *SEDIMENT-TRANSPORT CURVES*.

Hudson, N. W. (1997). Medición sobre el Terreno de la Erosión del Suelo y de la Escorrentía. (F. & A. Org., Ed.) (Vol. 68).

Iroumé, A. (1990). Assessment of Runoff and Suspended Sediment Yield in a Partially Forested Catchment in Southern Chile. *Water Resources Research*, 26(11), 2637–2642. <https://doi.org/10.1029/WR026i011p02637>

Meyer-Peter, E., & Müller, R. (1948). Formulas for bed-load transport. In *IAHSR 2nd meeting, Stockholm, appendix 2*. IAHR.

Parker, G., Klingeman, P. C., & McLean, D. G. (1982). Bedload and size distribution in paved gravel-bed streams. *Journal of the Hydraulics Division*, 108(4), 544–571.

Smart, G. M. (1984). Sediment transport formula for steep channels. *Journal of Hydraulic Engineering*, 110(3), 267–276.

OPTIMIZACIÓN DE LA TÉCNICA DEL RIEGO PASIVO PARA EL CONTROL DE LOS EFECTOS DE LAS HELADAS POR MODELAMIENTO Y SIMULACIÓN

**Optimization of the passive irrigation technique to control the effects of frost events by modeling and simulation*

Diego Renzo Ramírez Chávez

Universidad Nacional del Centro del Perú
Facultad de Ingeniería Química
diego.rch.x@gmail.com

Asesora: Dra. Yamina Silva Vidal

Subdirección de Ciencias de la Atmósfera e
Hidrosfera - IGP
Tesis desarrollada en el Observatorio de Huancayo

RESUMEN

Se realizaron seis experimentos en una parcela ubicada en el Observatorio de Huancayo del IGP en Huayao (distrito de Huachac, provincia de Chupaca, Junín), con el objetivo de evaluar la efectividad del riego como técnica de mitigación de los efectos de las heladas en los cultivos. Para ello se evaluó el comportamiento de los incrementos del contenido volumétrico de agua en el suelo, a diferentes alturas de agua aplicada en la técnica de riego pasivo. De igual forma se elaboró un modelo matemático a partir del balance energético en la superficie, el cual fue validado utilizando datos diarios de los meses donde se registraron mayor frecuencia de heladas en el año 2016. Los datos experimentales obtenidos sobre el contenido volumétrico de agua en el suelo y el modelo validado fueron integrados con el fin de determinar el comportamiento de la temperatura de superficie para un horario nocturno, lo que es parte fundamental en la evaluación de la técnica del riego pasivo. A partir de los resultados obtenidos se concluye que el modelo tiene una buena estimación de la temperatura de superficie para el horario determinado y que a partir del modelo el valor óptimo de la altura de agua que se debe aplicar es de 25 mm, el cual permite obtener un incremento de la temperatura mínima de superficie de 2.23°C.

Palabras clave: Contenido volumétrico de agua, técnica del riego, helada, modelo, Valle del Mantaro.

ABSTRACT

Six experiments were conducted on a parcel located at the IGP Huancayo Observatory in Huayao (Huachac district, Chupaca province, Junín), to evaluate the effectiveness of irrigation as a technique to mitigate the effects of frost on the crops. For this, the behavior of increases in the volumetric content of water in the soil was evaluated at different heights of water applied in the passive irrigation technique. In the same way, a mathematical model was elaborated from the energy balance on the surface, which was validated using daily data from the months with the highest frost frequency in 2016. The experimental data obtained on the volumetric content of water in the soil and the validated model were integrated in order to determine the behavior of the surface temperature for a night time, which is a fundamental part in the evaluation of the passive irrigation technique. From the obtained results, we conclude that the model has a good estimate of the surface temperature for the determined schedule, and the optimal amount of water height to be applied is 25 mm, which allows obtain an increase in the minimum surface temperature of 2.23 °C.

Keywords: Volumetric water content, irrigation technique, frost, model, Mantaro Valley.

1. INTRODUCCIÓN

Los procesos de transferencia de energía entre el suelo y la atmósfera han sido estudiados ampliamente en el extranjero; sin embargo, en nuestro país y específicamente en nuestra región, existen escasos estudios dedicados a este tema, siendo Saavedra (2013) el pionero en estudiar los procesos radiativos asociados a las heladas en el Valle del Mantaro.

El fenómeno meteorológico que causa mayor daño y de mayor frecuencia en el Valle del Mantaro es la helada (Trasmonte, 2009; Trasmonte y Enciso, 2012); que a pesar de ser conocido, hay poca información sobre la mitigación de los impactos y medidas de protección de cultivos contra este fenómeno. Ventskevich (1958) nos ayuda a definir el fenómeno de helada como "Cuando la temperatura de la superficie cae por debajo de 0°C"; y Cunha (1952) como "La existencia de una temperatura baja del aire que causa el daño o muerte de las plantas, sin mencionar la formación de hielos".

Una de las técnicas de mitigación de efectos de las heladas que se usa actualmente en el Valle del Mantaro es la técnica del riego pasivo que cuenta con muy poco estudio y más bien es conocimiento empírico. La FAO (2017) (Food and Agriculture Organization of the United Nations) la define como "Humedecer los suelos secos con antelación a la helada, de forma que el Sol pueda calentarlo y así poder mitigar los efectos de las heladas".

2. ZONA DE ESTUDIO Y METODOLOGÍA

2.1. Zona de estudio

La investigación se desarrolló en el Observatorio de Huancayo del Instituto Geofísico del Perú (IGP), ubicado en el departamento de Junín, provincia de Chupaca, distrito de Huachac, muy cercano a la población de Huayao. Se muestra su ubicación geográfica en la Figura 1.

Sus coordenadas geográficas corresponden a 12° 02' 26.59" S 75° 19' 14.11" O y fue fundado el 1 de marzo de 1922.

El periodo en el que se realizó las experimentaciones estuvo entre junio a setiembre del año 2017.



Figura 1. Ubicación geográfica de la investigación
Fuente: Google Earth 2017

2.2. Diseño experimental

Se realizaron seis experimentaciones en una parcela experimental de 4m², donde se ubican los instrumentos necesarios para la investigación. Los periodos de medición fueron de 11 horas, iniciando a las 19:00 horas (hora local) y finalizando a las 06:00 horas del día posterior. Se registraron la temperatura de superficie, radiación de onda larga incidente, el perfil de temperatura del suelo hasta los 50 cm de profundidad, humedad del suelo, velocidad del viento, temperatura ambiente y la nubosidad presente. Todos los datos son registrados con una frecuencia de

1 min, a excepción del perfil de temperatura y la humedad, los cuales están a frecuencia de 5 min.

Todos los experimentos se llevaron a cabo en el periodo típico de heladas en el Valle del Mantaro, a excepción del último experimento que se llevó a cabo en setiembre, tal como se muestra en la Tabla 1. Para el Valle del Mantaro el periodo de heladas fue caracterizado por Trasmonte y Enciso (2012) y contrastados en la presente investigación. Este periodo abarca los meses de junio, julio y agosto.

N°	Fecha	Altura de agua aplicada, mm	Volumen de agua aplicada, L
1	05/06/2017	5	20
2	13/06/2017	10	40
3	20/06/2017	20	80
4	10/07/2017	25	100
5	19/07/2017	35	140
6	05/09/2017	50	200

Tabla 1. Diseño experimental de la investigación
Fuente: Elaboración Propia

El riego usado para la aplicación de la técnica, es la que comúnmente se usa en todo el Valle del Mantaro, riego por gravedad, que es el método tradicional de riego con un flujo de agua continuo en los terrenos agrícolas.

2.3. Modelo de difusión de energía

En el límite superior (superficie) sucede la mayor transferencia de energía por la interacción con la atmósfera que genera diversos procesos físicos. La magnitud de estos va a depender del tiempo, generalmente si es en horario diurno o nocturno. Por ejemplo, la radiación de onda corta (radiación solar) es nula por las noche, pero en horario diurno alcanza sus máximos valores; en cambio la radiación de onda larga (infrarroja) tiene predominancia en horario nocturno generando así transferencia de energía.

Entonces, tenemos diferentes procesos que determinan la transferencia de energía, que además tiene diferencias entre el día y la noche (Figura 2) que se describen a continuación:

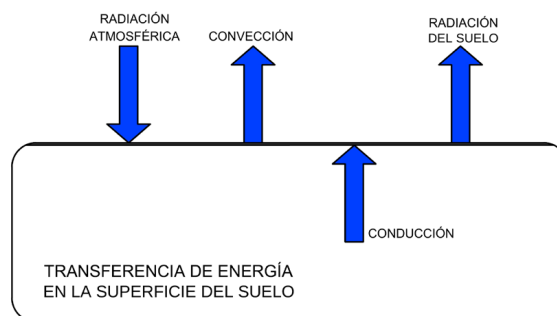


Figura 2. Balance energético sobre la superficie del suelo en horario nocturno.

Fuente: Elaboración propia

De manera general el balance energético en la superficie está dada por:

$$R_i + K = H + R_r$$

R_i = Radiación incidente en la superficie
 K = Energía por conducción
 H = Energía por convección
 R_r = Radiación reflejada.

Los procesos de transferencia de energía son descritos de manera matemática:

Transferencia de energía atmosférica por radiación

$$\text{Transferencia de energía por Radiación atmosférica} = R_{in} * a$$

Donde:
 a = Absortividad propia del suelo

Transferencia de energía por conducción

$$\text{Transferencia de energía por Conducción} = -k \frac{dT}{dy}$$

Donde:
 k = Constante de conductividad térmica expresada en $[\frac{W}{m \cdot ^\circ C}]$
 y = Profundidad, m.

Transferencia de energía por convección

$$\text{Transferencia de energía por Convección} = h * (T - T_{\infty})$$

Donde:
 h = Coeficiente de transferencia de energía por convección,
 T_{∞} = Temperatura ambiente, $^\circ C$

Transferencia de energía por radiación del suelo

$$\text{Transferencia de energía por Radiación del suelo} = \epsilon \sigma T^4$$

Donde:
 σ = Constante de Boltzman, $5.67 * 10^{-8} \frac{W}{m^2 * K^4}$
 ϵ = Emisividad

La validación estadística del modelo sigue la metodología planteada por Medina-Peralta, et al (2010), donde se usan los siguientes estadísticos de dispersión:

Eficiencia de Modelado (MEF), establece la concordancia entre valores observados con valores predichos. Su valor ideal es 1 y si su valor fuese negativo, los valores simulados por el modelo son tan inexactos que sería mejor usar la media de estos.

$$MEF = 1 - \frac{\sum_{i=1}^n (y_i - z_i)^2}{\sum_{i=1}^n (y_i - \underline{y})^2}$$

Coeficiente de determinación de Modelo, indica la proporción de la variación total de los datos observados con los simulados. Su valor adecuado es 1, si es superior a 1 es un indicador de subestimación y si es mayor a 1 es indicador de sobrestimación sobre datos observados.

$$CD = \frac{\sum_{i=1}^n (y_i - \underline{y})^2}{\sum_{i=1}^n (z_i - \underline{y})^2}$$

Coeficiente de error, es una medida promedio relativa de diferencias absolutas que es expresada como relación de la media de valores observados. Su valor adecuado es 0, es muy usado para describir el ajuste del modelo a los datos observados.

$$C = \frac{\sum_{i=1}^n |y_i - z_i|}{n * \underline{y}}$$

3. RESULTADOS

Al modelo elaborado se le añadió relaciones entre contenido de agua volumétrica en el suelo (VWC) con diferentes parámetros, por ejemplo, con conductividad térmica y capacidad calorífica, que son mostrados en la Figura 3.

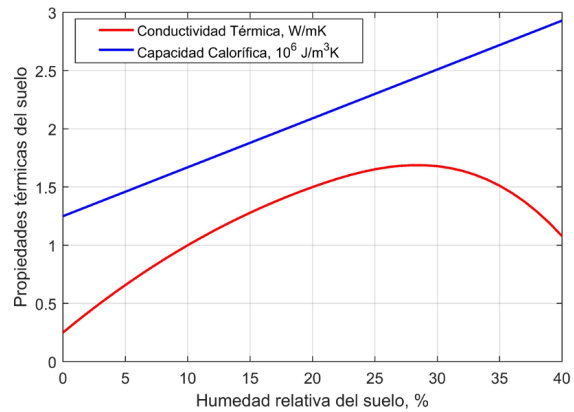


Figura 3.- Comportamiento de las propiedades térmicas del suelo en función al contenido volumétrico de agua. Fuente: Adaptado de (Geiger et al, 1995, p. 528)

La relación entre VWC y el albedo del suelo (Figura 4), indica que en un suelo con más contenido de agua, el albedo disminuye, por ende absorbe más energía solar en general.

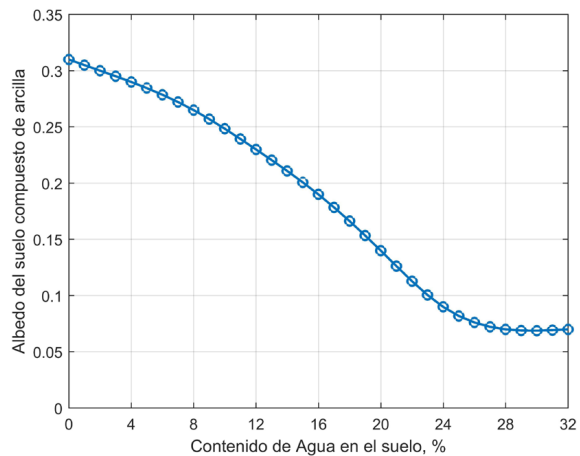


Figura 4.- Comportamiento del albedo en función al contenido de agua en el suelo. Fuente: Adaptado de (Idso, Jackson, & Reginato, 1975, p. 3)

La validación del modelo se realizó tomando fechas de ocurrencia de heladas en el año 2016, un total de 75. Al realizar

la validación se demostró que el modelo predice correctamente en 63 de las 75 pruebas realizadas, con un mayor porcentaje en el mes de julio. En la Figura 5 se presenta la comparación para el periodo de análisis, observando que el simulado tiene a sobre estimar los valores de la temperatura.

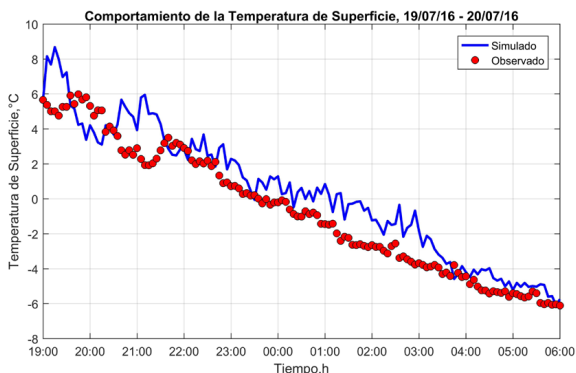


Figura 5.- Comparación entre datos observados (en rojo) y simulados (azul) de la temperatura de superficie para el periodo 19/06-20/06 de 2016.

Fuente: Elaboración propia.

En la parcela experimental se ubicaron instrumentos de medición como: tensiómetros de suelo, radiómetro infrarrojo, anemómetro, pirgeómetro de incidencia, termómetros de ambiente, que permiten registrar las variaciones del agua volumétrica en el suelo, VMC que se muestran en la Tabla 2.

N°	Fecha	Altura de Agua, mm	Incremento de VWC
1	05/06/2017	5	0.0121
2	13/06/2017	10	0.0143
3	20/06/2017	20	0.0228
4	10/07/2017	25	0.0247
5	19/07/2017	35	0.0255
6	05/09/2017	50	0.0261

Tabla 2.- Incrementos de VWC obtenidos mediante la experimentación.

Fuente: Elaboración propia.

En la Figura 6 se ha graficado la cantidad de agua incrementada en el suelo, en función a la altura de agua, donde puede observarse que luego de 0,025 se mantiene casi constante el volumen de agua.

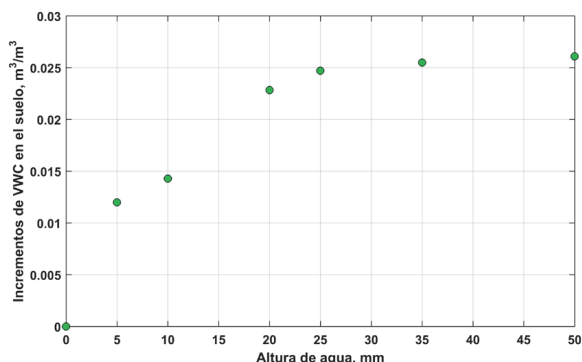


Figura 6.- Comportamiento de incrementos de VWC según altura de agua aplicadas en la técnica del riego.

Fuente: Elaboración propia

Los incrementos obtenidos fueron incorporados al modelo validado en un día aleatorio del año 2016, específicamente el día de helada 19/07/2018 donde se obtuvieron los incrementos de la temperatura de superficie en función a la altura de agua aplicada en la técnica del riego pasivo que son mostrados en la Figura 7.

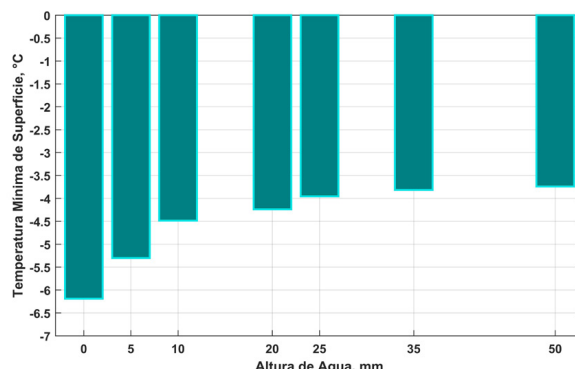


Figura 7.- Comportamiento de la temperatura mínima de superficie en función a la altura de agua aplica en la técnica del riego pasivo.

Fuente: Elaboración propia.

Un óptimo de la técnica del riego pasivo se considera utilizando 25 mm de altura de agua que eleva 2.23°C la temperatura de superficie, ya que el incremento posterior de la temperatura de superficie no es significativo por el motivo de no compensar el volumen de agua extra que debe usarse para el poco incremento de la temperatura de superficie.

4. DISCUSIÓN

La metodología usada en la presente investigación es similar a la planteada por Lhome & Vacher (2003) en su investigación “Mitigación de las heladas en los camellones del Altiplano andino”, que elabora un modelo y su uso para optimizar la técnica de camellones usado en Puno. Nuestra investigación aplica dicha metodología para elaborar el modelo y lo valida.

Por su parte, Saavedra (2013) elaboró un modelo numérico de difusión de energía con muy buena estimación de la temperatura de superficie, por nuestra parte se realizó un modelo similar, añadiendo transferencia por calor sensible y parámetros físicos que dependen del VWC que nos dan resultados que estiman más eficientemente los datos observados.

5. CONCLUSIONES

Luego de determinar que se demostró la hipótesis planteada y se alcanzaron los objetivos propuestos, se finaliza la investigación manifestando las siguientes conclusiones y recomendaciones:

A través del balance energético realizado para la superficie de un suelo desnudo de textura franco arcillosa arenosa, se elaboró un modelo matemático que describe la temperatura de superficie para un horario nocturno.

A través del modelo matemático elaborado y su posterior simulación, se determinó que la altura de agua óptima que se debe aplicar a la técnica del riego pasivo es 25 mm, lo que permite un incremento de 2.23°C en la temperatura mínima de superficie.

6. RECOMENDACIONES

Realizar un modelamiento del proceso físico teniendo en cuenta el perfil de contenido volumétrico de agua en el suelo y evaluar su comportamiento con respecto al modelo elaborado en la presente investigación.

Realizar una investigación utilizando sembríos propios del Valle de Mantaro, con el objetivo de determinar la eficiencia que tiene el modelo para describir la temperatura de superficie con la presencia de sembríos como papa, maíz, etc. De no ser eficiente el modelo, se puede añadir las variables propias de los sembríos para mejorar la eficiencia del modelo y determinar una "temperatura de cultivo" que pueda ser un indicador más eficiente para determinar el grado de los efectos de las heladas.

7. AGRADECIMIENTOS

El presente estudio fue financiado por el proyecto "Estudio de los procesos físicos que controlan los flujos superficiales de energía y agua para el modelado de heladas, lluvias intensas y evapotranspiración en la sierra central del Perú", Contrato N° 400-PNICP-PIBA-2014, financiado por Innóvate Perú.

Mis agradecimientos al Instituto Geofísico del Perú, por permitirme desarrollar esta investigación. A la Ph. D. Yamina Silva Vidal por su asesoría y la confianza puesta en mí para ser parte del proyecto y realizar la presente investigación. Así como a los ingenieros y licenciados del Instituto Geofísico del Perú por su ayuda incondicional, a Lucy Giráldez, Luis Suarez, Stephany Callañaupa, Miguel Saavedra y Jairo Valdivia por sus consejos y ayuda técnica para el presente estudio.

8. BIBLIOGRAFÍA

Cunha, J.M. (1952). Contribuição para o estudo do problema das geadas em Portugal. [in Portuguese] Relatório final do Curso de Engenheiro Agrônomo. I.S.A., Lisbon.

FAO. (2010). Protección contra las Heladas: Fundamentos, práctica y economía. Vol. I. Roma.

Geiger, R., R.H. Aron, and P. Todhunter (1995). The Climate near the Ground, 528 pp., ViewegVerlag, Braunschweig, Germany,

Idso, S., Jackson, R., Reginato, R., Kimball, B., & Nakayama, F. (1975). The dependence of bare soil albedo on soil water content. *Journal of applied meteorology*, 14, 109. doi: 10.1175/1520-0450(1975)014<0109:TDOBSA>2.0.CO;2.

Lhomme, Jean-Paul, Vacher Jean-Joinville. (2003). La mitigación de heladas en los camellones del altiplano andino. *Bulletin de l'Institut Français d'Etudes Andines*, 32 (2), 377-399.

Medina-Peralta, S, Vargas-Villamil, L, Navarro-Alberto, J, Canul-Pech, C, & Peraza-Romero, S. (2010). Comparación de medidas de desviación para validar modelos sin sesgo, sesgo constante o proporcional. *Universidad y ciencia*, 26(3), 255-263. Recuperado en 15 de marzo de 2020, de http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0186-29792010000300005&lng=es&tlng=es.

Saavedra, M. (2013). Caracterización física de heladas radiativas en el Valle del Mantaro (Tesis para optar el título profesional de Licenciado en Física). Universidad Nacional Mayor de San Marcos, Lima, Perú. Disponible en: <http://repositorio.igp.gob.pe/handle/IGP/964>

Trasmonte, G. y L. Enciso. (2012). Antecedentes generales sobre heladas en el Valle del Mantaro. Eventos meteorológicos extremos, En Libro Manejo de Riesgo ante Eventos Meteorológicos Extremos en el Valle del Mantaro, Vol.1. <http://repositorio.igp.gob.pe/handle/IGP/740>.

Trasmonte, G. L. (2009). Propuesta de gestión de riesgo de heladas, que afectan a la agricultura del Valle del Mantaro (Andes Centrales del Perú) (Tesis para optar el grado académico de Maestra en Ecología y Gestión Ambiental). Universidad Ricardo Palma, Lima, Perú.

Ventskevich, G.Z. (1958). Agrometeorology. Translated from the Russian by the Israel Programme for Scientific Translation, Jerusalem, 1961.



CIENCIAS DE LA TIERRA SÓLIDA

SECCIÓN 3

MODELOS DE DEFORMACIÓN CORTICAL DURANTE LA FASE INTERSÍSMICA Y SU COMPARACIÓN CON LAS VELOCIDADES GPS PARA LA ZONA DE SUBDUCCIÓN DEL PERÚ

**Crustal deformation models during the interseismic phase and their comparison with the GPS velocities in the Peruvian subduction zone*

Eliana Amelia Mantilla Calisaya

Universidad Nacional de San Agustín de Arequipa
Facultad de Geología, Geofísica y Minas
emantillac@gmail.com

Asesor: Dr. Juan Carlos Villegas Lanza

Unidad de Geodesia Espacial – Ciencias de la Tierra Sólida IGP

RESUMEN

En este estudio se han elaborado modelos de dislocación elástica para simular la deformación superficial de la corteza terrestre durante fase intersísmica y estimar la máxima profundidad del acoplamiento sísmico en la interfaz de contacto de las placas tectónicas de Nazca y Sudamericana en la zona de subducción del Perú. Para el modelamiento se utiliza la metodología de Okada (1985) y los parámetros geométricos de los planos de falla que caracterizan esta región. Estos modelos son validados con las velocidades intersísmicas GPS de todos los estudios realizados y publicados en el Perú hasta la fecha. Los resultados muestran que la profundidad de la interfaz de contacto de placas con alto nivel de acoplamiento sísmico, para la región norte (Lat. 3°S a Lat. 10°S) está comprendida entre los 5 y 20 km de profundidad, para la región centro (Lat. 10°S a Lat. 14°S) es más profunda y va desde los 5 hasta 45 km de profundidad y para la región sur (Lat. 14°S a Lat. 19°S) está comprendida entre los 5 y 43 km de profundidad. Los desplazamientos generados por el modelo son coherentes con los datos GPS observados.

Palabras clave: Deformación cortical, acoplamiento sísmico, zona sismogénica

ABSTRACT

In this study we have performed interseismic elastic dislocation models for the interseismic phase in order to estimate the surface crustal deformation and the maximum depth of locking along the plate interface in the Peruvian subduction zone. The modeling approach uses the Okada, (1985) methodology and parameters of the fault plane. These models are validated with information of published interseismic GPS velocities in Peru so far. The obtained results show that the depth of the plate interface that is highly locked, for the northern Peru segment (Lat. 3°S to Lat. 10°S) is comprised between 5 and 20 km depth, for the the central region (Lat. 10°S to Lat. 14°S) is deeper from 5 to 45 km depth, and for the southern segment (Lat. 14°S to Lat. 19°S) is comprised between 5 to 43 km depth. The surface GPS displacements are in good agreement with the observed GPS velocities.

Keywords: Crustal deformation, seismic coupling, seismogenic zone.

1. INTRODUCCIÓN

En el Perú la actividad sísmica está relacionada directamente con el proceso de subducción de la placa de Nazca bajo la placa Sudamericana, el cual se da a una velocidad relativa de 6-7 cm/año (Kendrick et al., 2003). Los rasgos estructurales, geomorfológicos, fallas geológicas y la deformación superficial continental es el resultado de dicho proceso. En la zona de subducción peruana se encuentran superficies en la interfaz de contacto que ofrecen fricción al libre deslizamiento y que generan acumulación de esfuerzos y energía sísmica concentrándose en la zona sismogénica, a este proceso se le denomina acoplamiento sísmico.

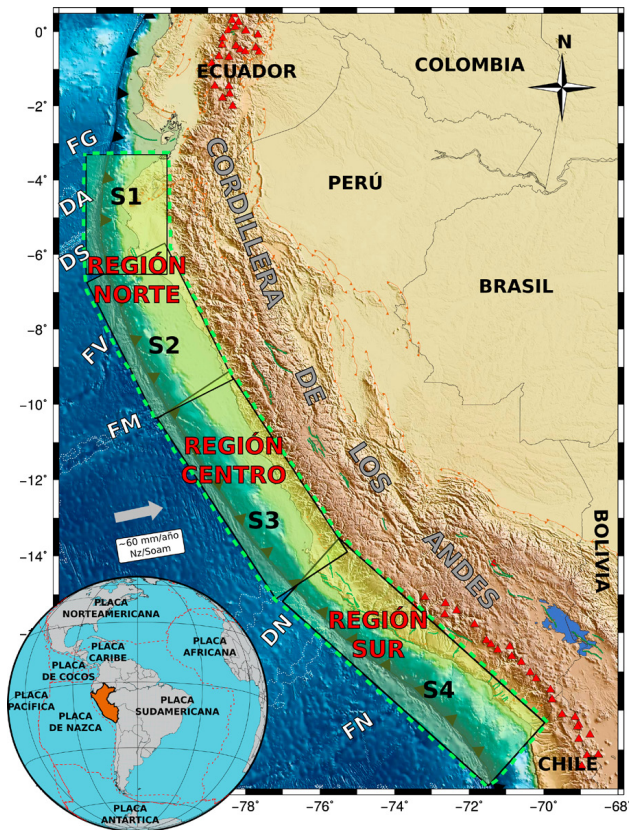


Figura 1: Mapa de ubicación del área de estudio. Los recuadros verdes de líneas discontinuas corresponden a los segmentos modelados en la zona de subducción del Perú. Fuente Gebco.

En los últimos años el empleo de la técnica geodésica GPS/GNSS ha permitido identificar y cuantificar la deformación cortical generada por el acoplamiento sísmico. En el Perú esta técnica viene siendo empleada desde 1998 y pone en evidencia desplazamientos cosísmicos y la deformación intersísmica que a su vez ha servido para elaborar modelos que muestran las zonas acopladas a lo largo del borde occidental del Perú [Norabuena et al., 1998; Kendrick et al., 2001; Chlieh et al., 2011; Nocquet et al., 2014; Gagnon et al., (2015); Villegas-Lanza et al., 2016 y Quiroz, 2016].

El presente estudio tiene como objetivo elaborar modelos de dislocación elástica durante la fase intersísmica para simular la deformación superficial de la corteza terrestre y estimar la profundidad máxima de acoplamiento sísmico en la zona de contacto Nazca/Sudamérica de la zona de subducción del Perú (Fig. 1).

2. CICLO SÍSMICO

Este concepto fue introducido por Reid en 1910 luego del gran terremoto de San Francisco, dando lugar a la teoría del rebote

elástico, la cual permite explicar los procesos de acumulación y liberación de esfuerzos y deformación que ocurren entre dos terremotos que presentan las mismas características.

Este ciclo está conformado por cuatro fases: la fase intersísmica, donde se produce la acumulación continua de los esfuerzos por décadas a cientos de años e induce deformación superficial de la corteza terrestre. La fase presísmica, muy poco observada, se distingue por la ocurrencia de sismos precursoros y deformación transitoria que puede durar algunos años, meses, días, horas o minutos antes del terremoto. La fase cosísmica es cuando de forma súbita se produce la liberación de los esfuerzos acumulados en segundos a minutos, dando lugar al terremoto. Por último, la fase postsísmica se inicia después del terremoto y presenta procesos de relajación elástica, réplicas y deslizamientos asísmicos en un tiempo de meses, años, etc. hasta que la interfaz de contacto de la falla entra nuevamente en la etapa intersísmica (Fig.2).

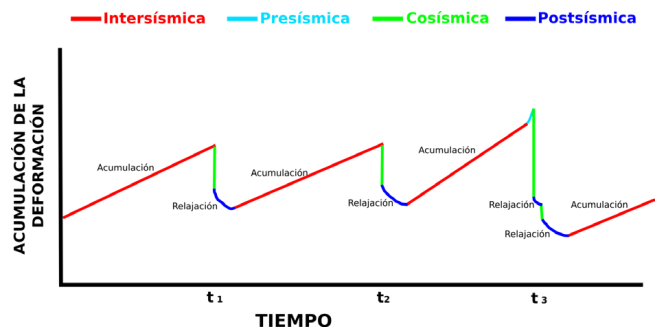


Figura 2: Esquema de las fases del ciclo sísmico.

3. SISTEMA GPS/GNSS

El sistema GPS/GNSS proporciona la posición relativa de un objeto en cualquier momento y lugar sobre la superficie terrestre, basada en el principio de la trilateración satelital, el cual requiere como mínimo la visualización de 4 satélites para definir la posición que está dada por la latitud, longitud, altura y el tiempo.

Este sistema está estructurado por tres segmentos: el segmento espacial, el segmento control y el segmento usuario (Fig.3). En el campo de la investigación en geociencias los sistemas GNSS contribuyen al monitoreo y cuantificación de la deformación cortical en fallas activas, zonas de subducción, deformación volcánica, glaciares, ionósfera, entre otros.

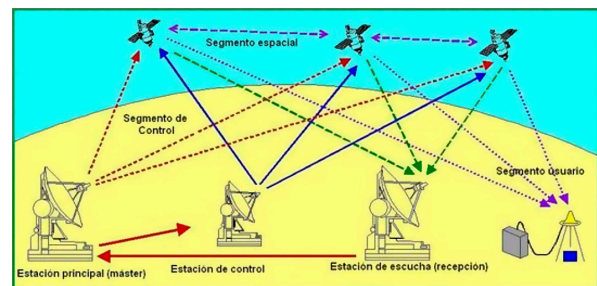


Figura 2: Esquema de las fases del ciclo sísmico.

En el Perú las primeras observaciones GPS fueron realizadas en 1994 en el marco del proyecto SNAPP (Norabuena et al., 1998). Dicho estudio mostró la contribución simultánea del bloqueo en la interfaz de placas y la deformación cortical en las regiones centro y sur del Perú.

Gagnon et al., (2005) combinaron mediciones geodésicas del fondo marino y de la superficie terrestre en región central del Perú, estimando una zona acoplada entre los 2 y 40 km de

profundidad. Luego Chlieh et al., (2011), recopilaron todos los datos geodésicos reportados anteriormente y estimaron el acoplamiento sísmico para las regiones centro y sur del Perú.

Posteriormente, Nocquet et al., (2014) realizaron observaciones GPS y estimaron el acoplamiento sísmico para las regiones centro y norte del Perú, además reportaron la presencia de un bloque tectónico denominado Silver Perú en la región norte. Luego Villegas-Lanza et al., (2016) obtuvieron el campo de velocidades GPS intersísmicas y determinaron el acoplamiento sísmico para toda la zona de subducción peruana y delimitaron el bloque Silver Perú. Finalmente, Quiroz, (2016) obtuvo resultados preliminares de los desplazamientos cosísmicos ocasionados por el sismo de Acarí (7.0Mw) del 2013 para la región sur del Perú.

4. ACOPLAMIENTO SÍSMICO

Se refiere al proceso en el cual la zona de contacto de una falla se encuentra bloqueada debido a la presencia de superficies que ofrecen resistencia al libre desplazamiento localizadas en la superficie deslizante, originando una acumulación de esfuerzos en la zona de contacto y deformación en la superficie terrestre.

4.1. Metodología de Okada

Describe un conjunto de expresiones analíticas que representan la acumulación y relajación de los esfuerzos basados en el modelo back-slip (Savage, 1983).

Esta metodología permite estimar desplazamientos superficiales u_x, u_y, u_z para una fuente rectangular debido a una dislocación elástica U_1, U_2 y U_3 (Fig.4B) en un semiespacio homogéneo e isotrópico. La fuente está caracterizada por un ancho W , un largo L , a una profundidad z , con un ángulo de inclinación δ , un ángulo strike ϕ y que experimenta un deslizamiento con un ángulo rake λ (Fig.4A). Finalmente Okada resume las ecuaciones matemáticas en la siguiente expresión:

$$u_x, u_y, u_z = F_{Okada}(x, y, d, \phi, \delta, L, W, \lambda, \underline{D})$$

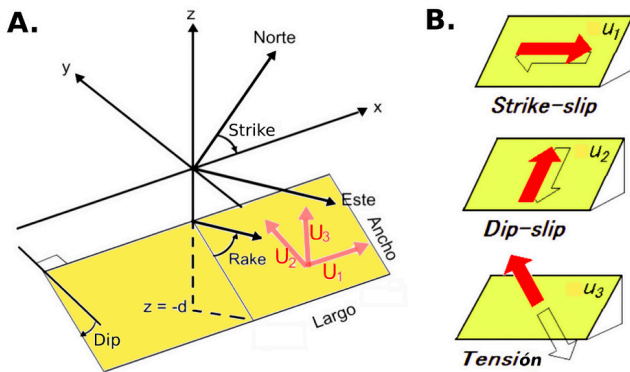


Figura 4: Representación gráfica del plano de falla para Okada, (1985).

5. MODELOS DE DISLOCACIÓN ELÁSTICA

Para un mejor desarrollo de los modelos se ha dividido la zona de estudio en cuatro segmentos S1, S2, S3 y S4 (Fig.1) y se han obtenido los parámetros geométricos del plano de falla para cada uno a partir de la tendencia de la sismicidad, los parámetros de los mecanismos focales de fallamiento inverso, los modelos de la

superficie de la placa de Nazca y otros estudios relacionados con dichos parámetros.

5.1. Parámetros del plano falla

Mediante Okada, (1985) se evalúan y definen los parámetros geométricos del plano de falla en cada segmento, realizando modelos iniciales para su respectivo ajuste mediante pruebas de ensayo y error (Tabla 1).

Región	Segmento	Strike ϕ (°)	Dip δ (°)	Rake λ (°)	Largo L (km)
Norte	S1	5	12	90	400
	S2	334	12	80	480
Centro	S3	324	18	78	600
Sur	S4	311	20	50	770

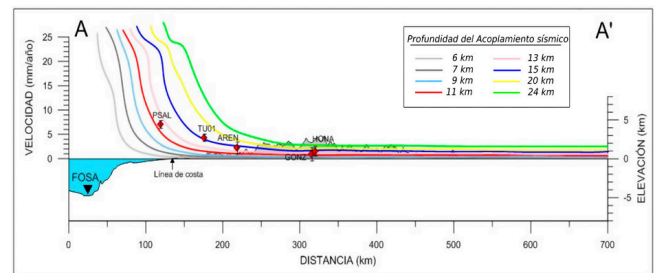
Tabla 1: Parámetros geométricos ajustados para los 4 segmentos.

5.2. Perfiles del gradiente de velocidad

Luego de definir dichos parámetros se elabora nuevos modelos para evaluar la profundidad máxima de la zona acoplada. Para lo cual se elaboraron perfiles de gradiente de velocidad horizontal los cuales son comparados con las proyecciones horizontales de las velocidades intersísmicas GPS (Fig.5). En este estudio se han unificado y expresado en un único marco de referencia todas las velocidades GPS (vector rojo) de los estudios publicados a la fecha, lo cual permite contar con velocidades GPS de 121 estaciones distribuidas a lo largo del borde occidental del Perú (Fig.6).

El perfil A-A' del segmento S1, muestra que las curvas de profundidad a 11 y 13 km se ajustan a las velocidades GPS. El perfil F-F' del segmento S2 las curvas de 15 y 20 km se ajustan más a las proyecciones de velocidades GPS. Estos dos perfiles permiten caracterizar la profundidad de acoplamiento máximo en la región norte del Perú (Fig.5).

El perfil J-J' del segmento S3 de la región central del Perú, las curvas de 39 y 45 km de profundidad se aproximan mejor a las velocidades GPS, definiendo un acoplamiento más profundo. Por último, el perfil O-O' del segmento S4 de la región sur del Perú, las curvas de profundidad 36 y 43 km se ajustan a las velocidades GPS observadas (Fig.5).



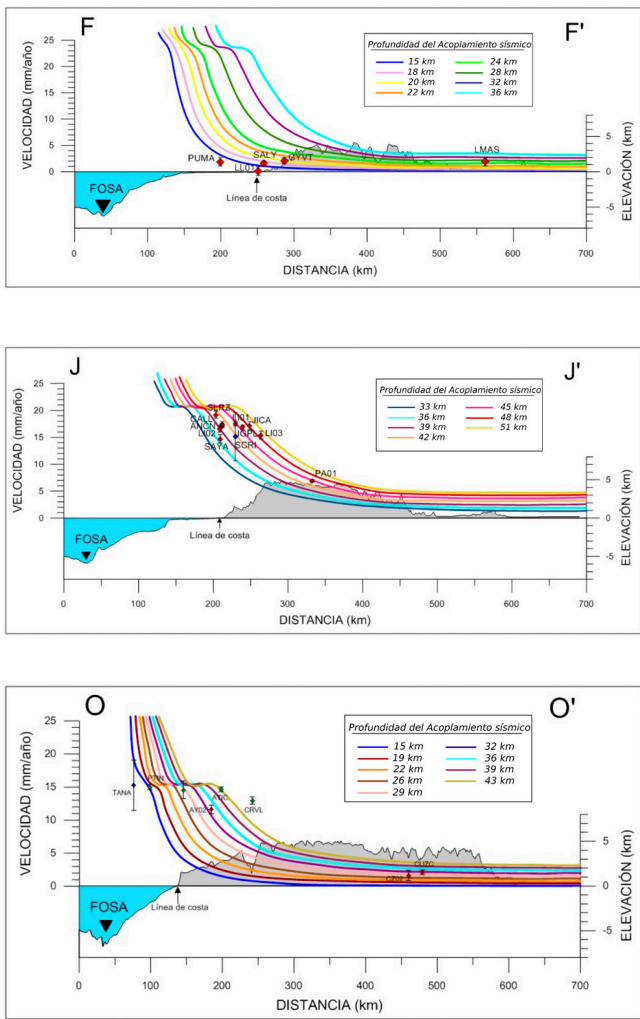


Figura 5: Perfiles del gradiente de velocidad horizontal de los modelos y las proyecciones horizontales de las velocidades GPS observadas para la zona de subducción del Perú.

6. RESULTADOS

Finalmente, para cada segmento se seleccionó el modelo que mejor se ajusta a las velocidades GPS observadas y se unifica en un modelo general, obteniendo desplazamientos modelados (vector azul) para todo el borde occidental del Perú (Fig.6).

Región	Segmento	Velocidad horizontal GPS observada (mm/año)		Desplazamiento horizontal modelado (mm)		Profundidad de la zona acoplada (km)
		V _{máx}	V _{mín}	D _{máx}	D _{mín}	
Norte	S1	7.7	0.4	8.7	0.8	5 a 11-13
	S2	3.6	0.2	3.1	0.9	5 a 15-20
Centro	S3	44.1	1.6	46.1	1.0	5 a 33-45
Sur	S4	23.9	1.6	24.3	1.8	5 a 36-43

Tabla 2: Desplazamientos superficiales modelados, velocidades GPS observadas y la profundidad del acoplamiento sísmico para los 4 segmentos.

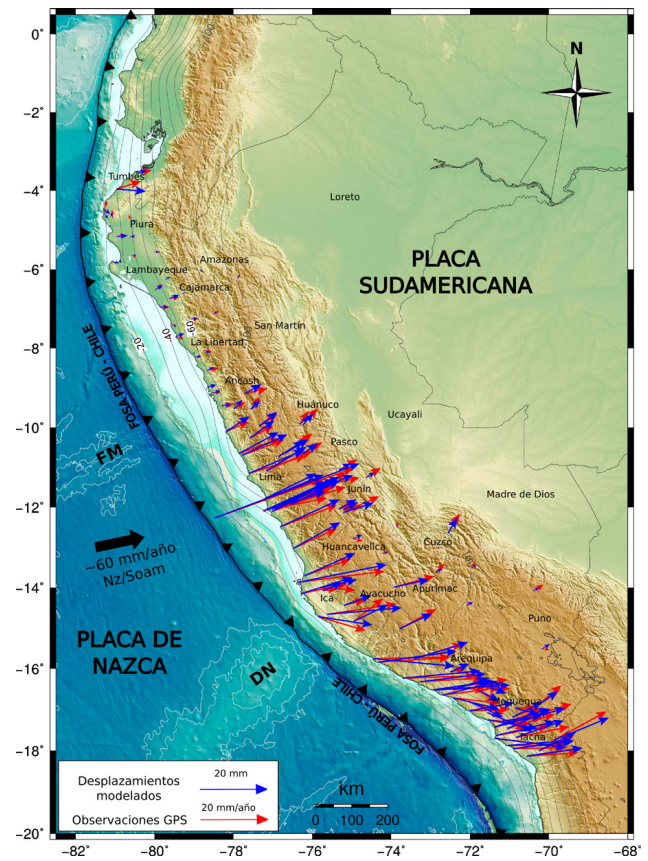


Figura 6: Mapa de desplazamientos superficiales modelados (vector azul) y las velocidades intersísmicas GPS (vector rojo) para el borde occidental del Perú.

7. DISCUSIÓN

Para el modelamiento de la acumulación de esfuerzos en zonas de subducción existe una diversidad de metodologías que incluyen geometrías y parámetros que reflejan las condiciones reales de una zona en particular, sin embargo el proceso de subducción en sí es complejo y requiere un conocimiento profundo y multiparamétrico. No obstante, la metodología de Okada, (1985) que utiliza una fuente de falla rectangular simple, permite simular muy bien la deformación superficial y la estimación del acoplamiento sísmico en zonas de subducción o fallas activas.

8. CONCLUSIONES

El análisis de los resultados de los modelos de dislocación elástica obtenidos, permite concluir lo siguiente:

Para la zona de subducción de la región norte (S1 y S2) se ha estimado un acoplamiento sísmico comprendido entre los 5 y 13-20 km de profundidad consistente con desplazamientos superficiales que varían entre 0.8 y 8.7 mm. Esta región se caracteriza por la ausencia de grandes sismos ($\geq 8.0Mw$), ninguno en los últimos 500 años, sin embargo, los modelos sugieren que el acoplamiento sísmico superficial podría generar sismos de magnitud moderada ($\geq 7.0 Mw$) y/o terremotos-tsunamis.

Para la región central del Perú (S3) se ha estimado una profundidad máxima de acoplamiento sísmico de 45 km, el cual genera en superficie desplazamientos hacia E-NE con valores desde 46.1 mm para el punto SE-SW ubicado cerca de la fosa a 1.0 mm en el interior del continente. En esta región, a pesar de la ocurrencia de terremotos ($\geq 8.0Mw$) que cubren aproximadamente el área de ruptura del sismo de 1746, aún existe energía sísmica suficiente como para generar un sismo de gran magnitud [Chlieh et al., (2011); Villegas-Lanza et al., (2016)]. Los modelos sugieren que

la interfaz de placas está altamente acoplada y podría generar un sismo de 8.5Mw.

Para la región sur del Perú (S4) se estima un acoplamiento sísmico entre los 5 y 43 km de profundidad con desplazamientos superficiales entre 24.3 mm para el borde costero y 1.8 mm en el interior del continente. Esta región al igual que la central, es la más afectada por terremotos, siendo el último el ocurrido en 2001 (8.4Mw), sin embargo existen zonas de silencio sísmico entre Arica-Tacna y Acarí-Nazca (Villegas-Lanza et al., 2016) en las cuales se espera sismos de gran magnitud. Los resultados del modelo muestran que los esfuerzos continúan acumulándose en la interfaz de placas y se espera que en dichas zonas ocurran sismos de magnitud 8.0Mw.

9. RECOMENDACIONES

Una mayor densificación de estaciones GPS permanentes en las áreas costeras, a fin de obtener mayor detalle de la distribución del acoplamiento sísmico.

La instalación de puntos geodésicos en el mar para todas las regiones, en especial para el segmento S2 y S3, con el fin de mejorar la resolución del acoplamiento sísmico superficial.

La combinación del monitoreo sismológico y geodésico continuo, con análisis de información en tiempo real, lo cual puede fortalecer los sistemas de alerta temprana ante la ocurrencia de un terremoto.

10. AGRADECIMIENTOS

Al Instituto Geofísico del Perú por facilitar los recursos necesarios para la realización de la presente investigación, la cual se realizó en el marco del proyecto convenio 100-2015/FONDECYT.

Agradecer al Dr. Juan Carlos Villegas Lanza por estar a cargo de mi capacitación, asesoramiento y su aporte valioso en el presente estudio y al personal de la Unidad de Geodesia Espacial, quienes me guiaron en el desarrollo de la tesis.

11. BIBLIOGRAFÍA

Chlieh et al., (2011). Interseismic coupling and seismic potential along the Central Andes subduction zone. *J. G. Res.*, 116(B12405).

Gagnon et al., (2005). Measuring the onset of locking in the Peru-Chile trench with GPS and acoustic measurements. *Nature*, 434(7030).

Kendrick et al., (2001). An integrated crustal velocity field for the central Andes. *Geochemistry Geophys. Geosystems*.

Nocquet et al., (2014). Motion of continental silver and creeping subduction in the northern Andes. *Nature Geoscience*, 287-291.

Norabuena et al., (1998). Space geodetic observations of Nazca-South America convergence across the Central Andes. *Science*, 279(5349).

Okada, (1985). Surface deformation due to shear and tensile faults in a halfspace. *Bull. of the Seism. Society of America*, vol. 75.

Quiroz, (2016). Interseismic velocity field in southern Peru: 4 years of GPS measures. *Université J. Fourier, Grenoble, France*.

Reid (1910). *The mechanics of the earthquake*. Carnegie Inst. of Washington.

Savage, J. C. (1983). A dislocation model of strain accumulation and release at a subduction zone. *Journal of Geophysical Research*, vol. 88.

Villegas-Lanza et al., (2016). Active tectonics of Peru: Heterogeneous interseismic coupling along the Nazca megathrust, rigid motion of the Peruvian Silver and Subandean shortening accommodation. *J. G. Res.*, vol. 121.

GEOFÍSICA Y SOCIEDAD

MANEJO Y GESTIÓN AMBIENTAL EN EL ECOSISTEMA MANGLAR: ESTUDIO COMPARATIVO ENTRE PERÚ - ECUADOR

**Environmental management in the mangrove ecosystem: Peru – Ecuador comparison*

Ing. Eduard Alder Feijoo Fox

Universidad Nacional de Tumbes
Escuela de Posgrado – Maestría en Ciencias
con mención en Gestión Ambiental
alder_1706@hotmail.com

Asesores: Mag. Luis Alberto Bermejo Requena – Universidad Nacional de Tumbes

Mag. Alejandra Graciela Martínez Grimaldo –
Instituto Geofísico del Perú

RESUMEN

Una característica en común de Perú (regiones de Piura y Tumbes) y Ecuador (provincias de Esmeraldas, Manabí, Santa Elena, Guayas y El Oro) es que comparten el ecosistema manglar, apenas separado por el río Zarumilla. Este ecosistema ofrece bienes y servicios ecosistémicos como captura de carbono, protección de las zonas costeras, entre otros; es hábitat de cuantiosas especies de aves, mamíferos, reptiles, peces, crustáceos, moluscos e insectos, y en él se realizan actividades como la extracción, recolección y pesca de crustáceos, bivalvos y peces. Ambos países han sufrido alteraciones en su superficie y calidad ambiental, principalmente por la instalación y la presencia del sector acuícola y expansión de las áreas urbanas (Perú y Ecuador), pero también por la palmicultura (Ecuador), factores que en las últimas cuatro décadas han generado una fuerte disminución de área del ecosistema. Los esfuerzos para la conservación del manglar han sido diferenciados entre Perú y Ecuador, por lo que este estudio busca analizar el manejo y la gestión ambiental del ecosistema en ambos países, identificando sus semejanzas y diferencias, así como haciendo un análisis crítico sobre sus respectivos avances y problemas.

Palabras clave: Manglares, Gestión Ambiental del Territorio, Perú, Ecuador.

ABSTRACT

A common feature of Peru (Piura and Tumbes regions) and Ecuador (provinces of Esmeraldas, Manabí, Santa Elena, Guayas and El Oro) is that they share the mangrove ecosystem, barely separated by the Zarumilla river. This ecosystem offers ecosystem goods and services such as carbon sequestration, protection of coastal zones, among others; it is the habitat of numerous species of birds, mammals, reptiles, fishes, crustaceans, mollusks, and insects; and it carries out activities such as the extraction, collection, and fishing of crustaceans, bivalves, and fishes. Both countries have suffered alterations in their surface and environmental quality, mainly due to the installation and presence of the aquaculture sector and the expansion of urban areas (Peru and Ecuador), but also by the palm farming (Ecuador), factors that in the last four decades have generated a strong decrease of the ecosystem area. The efforts for the conservation of the mangrove have been differentiated between Peru and Ecuador, so this study seeks to analyze the environmental management of the ecosystem in both countries, identifying their similarities and differences, as well as making a critical analysis of their respective advances and problems.

Keywords: Mangroves, Environmental Management of the Territory, Peru, Ecuador.

1. INTRODUCCIÓN

La gestión ambiental parte de la necesidad de la sociedad de conservar y mejorar la oferta y calidad ambiental, es decir, de los recursos que sirven para satisfacer las necesidades de los seres humanos y que son fundamentales como soporte de la vida en el planeta.

El manglar es uno de los ecosistemas que ha soportado mayor presión antropogénica, sobre todo en las últimas décadas, tanto por el desarrollo de la acuicultura, que necesita de grandes extensiones de terreno para la construcción de sus pozas (Bravo, 2010), pero también por el desarrollo urbano, la sobreexplotación de sus recursos hidrobiológicos, entre otros. Por ello, es un reto el detener y trabajar por la recuperación del ecosistema, con el fin de preservar y mejorar su calidad para las futuras generaciones. Un primer paso es el conocimiento de la gestión de la que ha sido objeto dicho ecosistema, con el fin de contribuir con información que sea útil para futuras políticas, planes, programas y proyectos.



Figura 1: Vista del ecosistema manglar. Golfo de Guayaquil, Ecuador. Crédito: P. Rengel Vela

Dado que un ecosistema es un medio continuo, no separado por barreras políticas, es importante hacer un análisis comparativo de la forma en que Perú y Ecuador han venido manejando el ecosistema, el cual ha variado de acuerdo a las condicionantes de orden político, económico y social particulares a cada país; además de considerarse que la coordinación de la temática ambiental entre ambos países podría ser crítica para la conservación de dicho ecosistema.

2. DESARROLLO

2.1. Características del ecosistema manglar en Perú y Ecuador

El manglar en ambos países tiene características biológicas similares, con la presencia de la concha negra (*Anadara tuberculosa*) y el cangrejo del manglar (*Ucides occidentalis*) como principales productos hidrobiológicos. Por otro lado, el manglar en Ecuador cuenta con siete especies de mangle: *Rhizophora mangle*, *Rhizophora harrisonii*, *Avicennia germinans*, *Conocarpus erectus*, *Laguncularia racemosa*, *Pelliciera rhizophorae* y *Mora megistosperma*, mientras que Perú solo cuenta con las cinco primeras, constituyendo el último relicto de este ecosistema en Sudamérica.

En cuanto a la superficie del ecosistema manglar en Ecuador y Perú ha tenido importantes variaciones en las últimas décadas. Para el caso de Ecuador, en 1969 la cobertura manglar era de 202 201 ha, sin embargo para el año 2006 la cobertura había disminuido a 148 230.23 ha (Clirsen, 2007), ubicadas en las provincias de Esmeraldas, Manabí, Santa Elena, Guayas y El

Oro. Mientras que en Perú la superficie del manglar en 1982 era de 5 964 ha, para 1992, el área se había reducido a 4 541 ha de bosques de mangle (ONERN 1992), mientras que para el año 2014 hubo un ligero incremento, identificándose 4 586.58 ha de manglar en Tumbes (Príncipe y Céspedes, 2014). El proceso de degradación acelerado pudo detenerse – al menos parcialmente –, gracias a la creación del área natural protegida Santuario Nacional Los Manglares de Tumbes SNLMT en 1988, el mismo que tiene una extensión de 2 972 ha. El resto de puntos importantes de manglar se ubican en Puerto Pizarro (Tumbes), y en San Pedro de Vice (Piura).

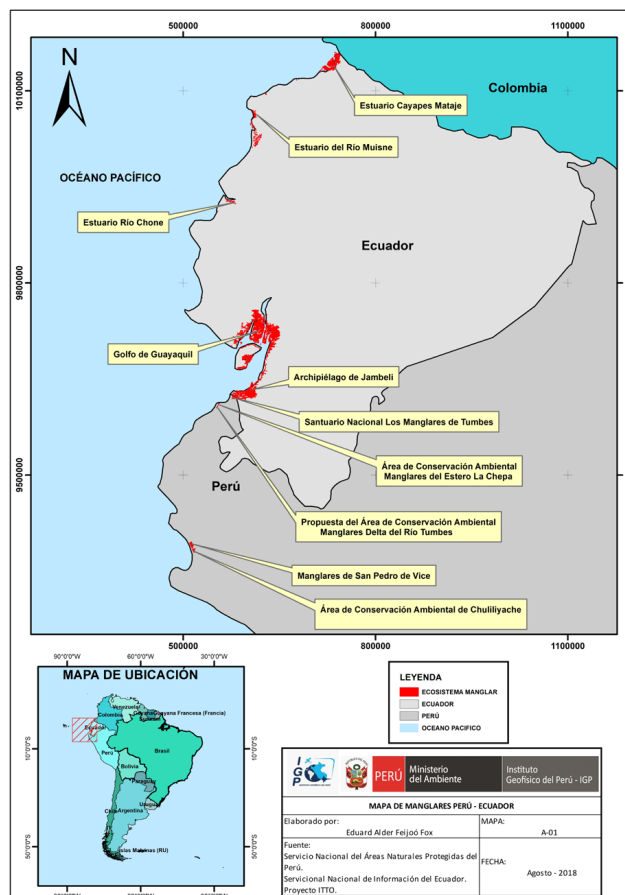


Figura 2: Principales áreas de manglares en Ecuador y Perú. Adaptado de Bravo (2014). Elaborado por A. Feijoo.

2.2. Causas de la degradación del ecosistema manglar

En Ecuador, la primera acción que degradó el ecosistema fue la conversión de manglares para el cultivo de cocoteros durante los años 40; posteriormente, muchas áreas fueron transformadas para ganadería y centros poblados (Bordero y Robadue, 1995). Finalmente, el cambio del uso del suelo para la construcción de camaronerías (llamadas langostineras en Perú) en la década del 70 tuvo un fuertísimo impacto, pues si bien las camaronerías se ubicaron primero sobre áreas de salinas o salitrales, posteriormente coparon zonas de manglar, en los años 80 esta actividad creció agresivamente, y para 1987, Ecuador ya era el primer exportador de camarón del mundo (Arévalo, 2014). Asimismo, en la costa ecuatoriana varios centros poblados y ciudades se han desarrollado sobre un aproximado de 5.000 ha de manglar (Bravo, 2010; Rodríguez y Espinoza, 2012).

a)



b)



Figura 3: Vista de una poza langostinera construida sobre el ecosistema manglar. Esmeraldas, Ecuador: P. Rengel Vela.

En Perú, los primeros esfuerzos para el desarrollo de la maricultura datan de fines de los años 50, sin embargo, es solo a inicios de los años 70 que se iniciaron los primeros experimentos sobre crianza, y que fueron base para la operación de las empresas privadas (citado por Alvarado, 1981; Del Solar, 1979; Peña 1970, en CDC - UNALM, 1986). Es solo a partir del año 1977 que la actividad langostinera se inicia con fuerza, inicialmente sobre áreas cubiertas por vegetación arbustiva y halófitas, pero entre los años 1980 -1987 se da una gran expansión en la zona manglar, talándose aproximadamente 600 ha en ese periodo (Inrena, 2007).

Además, el ecosistema manglar está afectado por las fluctuaciones cíclicas entre sequías y período de precipitaciones fuertes ocurridas durante el fenómeno El Niño, lo que resulta un avance y retroceso del manglar, especialmente en el lado peruano. Durante la sequía, ocurren problemas de hipersalinidad, generando la mortandad de rodales de mangle, mientras que las intensas lluvias lavan el exceso de sales y posibilitan el flujo de agua acarreado sedimentos y nutrientes (Ministerio de Agricultura, 2001).

Los años de eventos El Niño extraordinarios además pueden llegar a tener efectos adversos al manglar, como la elevación del nivel del agua muy por encima del nivel de marea alta, lo que ocasiona la muerte por asfixia de los árboles al quedar cubierta la zona de intercambio de gases. En condiciones normales debe ocurrir una regeneración del bosque, sin embargo, bajo las condiciones descritas los rodales muertos son finalmente talados y potencialmente convertidos en campos de cultivo (CDC

– UNALM, 1986), convergiendo factores naturales y antrópicas que pueden afectar negativamente al ecosistema.

2.3. Efectos de la degradación del ecosistema manglar

La degradación del ecosistema manglar a nivel internacional, ha mostrado tener importantes efectos negativos, entre los que se encuentran: la destrucción de hábitat de muchas especies, salinización excesiva de los suelos en las zonas de pozas para acuicultura abandonadas, salinización de acuíferos y contaminación de las aguas, erosión genética, afectación a la seguridad alimentaria, a la salud y a la protección de las comunidades costeras.

Adicionalmente, en Ecuador la degradación del ecosistema ha venido generando conflictos sociales, principalmente por el incumplimiento de las normativas legales que lo protegen el establece al ecosistema manglar como Patrimonio Forestal del Estado, Bosques Protectores, Patrimonio Nacional de Áreas, además de ecosistemas frágiles y Bienes Nacionales del Estado los cuales están fuera de comercio (C-CONDEM, 2007, citado en Rengel, 2013).

2.4. El ecosistema manglar y el cambio climático

El cambio climático es considerado como uno de los principales factores que actualmente amenazan la existencia de los manglares a nivel mundial, especialmente en su vinculación con el aumento del nivel del mar. Sin embargo, sus impactos son difíciles de medir (Flores, 2018) por su magnitud y complejidad, y por ende la causa más difícil de enfrentar. Si bien la sobreexplotación de los recursos y los desarrollos urbanísticos constituyen el eje central sobre el cual debe enfocarse la gestión ambiental de los ecosistemas de manglar a nivel local y regional (Uribe y Urrego, 2009), las implicancias del cambio climático para los manglares son de gran relevancia y deben ser conocidas por el gestor ambiental quien deberá tenerlas en cuenta para manejar adecuadamente estos ecosistemas y para emprender acciones de mitigación.

3. RESULTADOS PRELIMINARES

A pesar de compartir problemáticas similares, la gestión del ecosistema manglar ha tenido acciones diferenciadas para Ecuador y Perú, las que son resumidas a continuación.

3.1. Gestión ambiental del ecosistema en Ecuador

Desde los años 80, se empiezan a tomar en consideración los beneficios del ecosistema manglar para la sociedad, y en consecuencia se desarrollaron mecanismos para su conservación y protección (Coello et al., 2008), prohibiéndose su tala en 1985 a través del Decreto Ejecutivo 824-A. Posteriormente en 1986, mediante el Acuerdo Ministerial 498, se declara a los manglares como bosques protectores incluyendo no sólo a las partes cubiertas por bosques sino de "otras especies forestales y áreas salinas incluidas dentro del ecosistema" (Moreira, 2013).

Desde mediados de los años 90, empiezan en Ecuador las experiencias de conservación y uso sustentable del ecosistema manglar, otorgándoles un área de manglar a asociaciones privadas – inicialmente de extractores –, y posteriormente a comunidades ancestrales, con la finalidad de manejar sustentablemente los recursos sin atentar contra el bosque de manglar y sus recursos (Moreira, 2013). Hasta el año 2018, el Ministerio del Ambiente de Ecuador a través de la Dirección de Gestión y Coordinación Marina Costera tiene un total acumulado concesionado de 68002.93 ha de manglar a comunidades y grupos ancestrales de cinco provincias: Esmeraldas, Manabí, Santa Elena, Guayas

y El Oro, a través del Acuerdo de Uso Sustentable y Custodia del Manglar en Ecuador.

3.2. Gestión ambiental del ecosistema en Perú

Ante la pérdida acelerada del ecosistema, en 1988 se crea el área natural protegida "Santuario Nacional Los Manglares de Tumbes - SNLMT", fijando su superficie en 2972 ha, lo que representa más del 50% del total del ecosistema en el país, y que actualmente es administrada por el Servicio Nacional de Áreas Naturales Protegidas por el Estado - SERNANP, órgano adscrito al Ministerio del Ambiente. El resto del ecosistema ha sido -principalmente - parte de circuitos turísticos, lo que sin embargo no ha evitado su lenta degradación, por lo que actualmente se vienen trabajando en diversas estrategias de protección.

4. CONCLUSIONES PRELIMINARES

En ambos países el ecosistema manglar ha sufrido una considerable pérdida, debido principalmente a la actividad langostinera, pero también debido a otros factores antrópicos y ambientales. En la actualidad el ecosistema manglar es considerado como un sistema ecológico sumamente productivo debido a la gran diversidad biológica que alberga.

La adscripción a convenios internacionales como Ramsar, el Convenio de Diversidad Biológica, entre otros, ha permitido contar con un marco internacional para la protección del ecosistema.

En ambos países se han emitido normativas diferenciadas por cada país para la protección del ecosistema. Mientras que en Ecuador se ha generado normativa específica para el manglar, en Perú esa normativa es general, principalmente vinculada a la protección de ecosistemas costeros.

En Perú la protección del ecosistema se ha realizado a través de la creación de un Área Natural Protegida por el Estado (Santuario Nacional Los Manglares de Tumbes), mientras que en Ecuador la gobernanza del ecosistema manglar se realiza a través del uso sustentable y custodia del manglar, en el marco constitucional que le otorga derecho a la Naturaleza o Pacha Mama, que es donde se reproduce y realiza la vida, por lo que tiene derecho a que se respete integralmente su existencia y el mantenimiento y regeneración de sus ciclos vitales, estructura, funciones y procesos evolutivos.

5. RECOMENDACIONES

Es importante ahondar las investigaciones tanto sobre los aspectos normativos de la protección del ecosistema manglar, así como sobre la formación y desarrollo de las organizaciones sociopolíticas que usufructúan los recursos del ecosistema, y que son actores clave en su conservación.

6. AGRADECIMIENTOS

Deseo agradecer al Instituto Geofísico del Perú por facilitar los recursos necesarios para la realización de la presente investigación, la cual servirá para optar por el título de Magister en Ciencias con mención en Gestión Ambiental. Asimismo a mis asesores: Mag. Luis Bermejo y Mag. Alejandra G. Martínez por su apoyo, seguimiento y motivación constante.

7. BIBLIOGRAFÍA

Arevalo, N. 2014. Diagnóstico del Sector Camaronero en el Cantón El Guabo 2013". Trabajo de Titulación. Universidad Técnico de Machala.

Bordero, A., Robadue, D., 1995. Estrategia para el Manejo del Ecosistema Manglar en Ecuador.

Bravo, M., 2010. Interpretación del Estudio Multitemporal (Clirsen 1969-2006) de las coberturas de manglar, camaronerías y áreas salinas en la franja costera del Ecuador continental. Tesis para optar el grado en Maestría en Ciencias con énfasis en Manejo Sustentable de Recursos Bioacuáticos y Medio Ambiente. Universidad de Guayaquil.

Coello, S., Vinuesa, D., Alemán., 2008. Evaluación del desempeño de los acuerdos de uso sustentable y custodia de manglar de la zona costera del Ecuador. Ministerio del Ambiente de Ecuador. MAE.

CDC-UNALM, 1986. Estrategia de Conservación para los Manglares del Nor-Oeste peruano. Informe final del proyecto WWF-US. 3604. Lima – Perú.

Centro de Levantamientos Integrados de Recursos Naturales por Sensores Remotos, 2007. Actualización del estudio multitemporal de manglares, camaronerías y áreas salinas en la Costa Continental Ecuatoriana al año 2006.

Flores, D., 2018. Función de producción de la extracción de concha negra (*Anadara tuberculosa*) en el SNLMT, Perú. Tesis para optar el grado de Magister en Economía de Recursos Naturales y del Ambiente. UNALM.

Ministerio de Agricultura, 2001. Estrategia de conservación del ecosistema de los manglares de Tumbes – Perú 2001 – 2010.

Moreira, E., 2013. Evaluación del Estado Actual de las Concesiones de Áreas de Manglar Entregadas a Comunidades Ancestrales en la Provincia del Guayas. Tesis para optar el grado de Magister en Ciencias en énfasis en Manejo Sustentable de Recursos Bioacuáticos y el Medio Ambiente. Universidad de Guayaquil.

Instituto de Recursos Naturales, 2007. Plan Maestro del Santuario Nacional Los Manglares de Tumbes 2007 – 2011. Perú: Inrena.

Ministerio de Agricultura, 2001. Estrategia de conservación del ecosistema de los manglares de Tumbes – Perú 2001 – 2010.

ONERN., 1992. Monitoreo de los cambios en la cobertura y uso de la tierra en el ecosistema manglar de Tumbes (1982 - 1992). Volúmenes I y II. Convenio ONERN-ODC- FPCN. Programa de monitoreo de los manglares de Tumbes. Lima – Perú.

Príncipe, E., Céspedes., L., 2014. Análisis del territorio de la cobertura del manglar. En IGP (2014). Gestión sostenible de recursos naturales en el ecosistema de manglares de Tumbes: Análisis del territorio y diagnóstico socioeconómico.

Rengel, M., 2013. La Conservación del manglar, un debate político irresuelto: el caso de la propuesta de ley del manglar en el Ecuador (2008 – 2012). Tesis para optar el título de Magister en Ciencias Sociales con mención en Desarrollo Local y Territorial. Facultad Latinoamericana de Ciencias Sociales Sede Ecuador.

Rodríguez, M., Espinoza, G., 2012. Gestión Ambiental en América Latina y El Caribe: evolución, tendencias y principales prácticas. Banco Interamericano de Desarrollo.

Uribe, J., Urrego, L., 2009. Environmental management of mangrove ecosystems: An approach for the Colombian case.



Calle Badajoz 169,
Urb. Mayorazgo IV Etapa,
Ate, Lima, Perú
Central Telefónica: (511) 317 2300

<http://www.gob.pe/igp>

 <http://www.facebook.com/igp.peru>

 http://twitter.com/igp_peru

 https://www.youtube.com/c/igp_videos

