

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

MEJORAMIENTO DEL ENLACE DE COMUNICACIÓN INALÁMBRICA ENTRE UN DRON Y UNA ESTACIÓN TERRENA PROGRAMA EN MATLAB PARA PROCESAR LA INFORMACIÓN DE LA CALIDAD DEL ENLACE DE TELEMETRÍA DE UN UAV

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
TELECOMUNICACIONES**

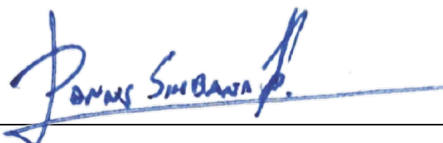
DANNY FERNANDO SIMBAÑA TASIGUANO
danny.simbana@epn.edu.ec

DIRECTOR: DIEGO JAVIER REINOSO CHISAGUANO
diego.reinoso@epn.edu.ec

DMQ, febrero 2023

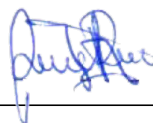
CERTIFICACIONES

Yo, Danny Fernando Simbaña Tasiguano declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



Danny Fernando Simbaña Tasiguano

Certifico que el presente trabajo de integración curricular fue desarrollado por Danny Fernando Simbaña Tasiguano, bajo mi supervisión.



Diego Javier Reinoso Chisaguano

DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Danny Fernando Simbaña Tasiguano

Diego Javier Reinoso Chisaguano

DEDICATORIA

Este trabajo se lo dedico a mi familia, quienes me han brindado su apoyo incondicional durante toda mi carrera universitaria y quienes han representado para mí, un ejemplo de esfuerzo y perseverancia.

Danny Simbaña

AGRADECIMIENTO

Agradezco a Dios

A mis padres Fernando y Sandra, por todo el apoyo, ánimos y consejos que he recibido durante toda mi vida, que me han permitido llegar hasta donde estoy ahora.

A mis amigos de la facultad con quienes nunca faltó el estudio, las risas y las penas. Y con quienes en conjunto hemos logrado llegar hasta el fin de esta carrera. De manera especial a Nubia a quien conozco desde casi el inicio, gracias por tu amistad.

A Mayerlin por motivarme a no darme por vencido, por siempre contagiarme de esas ganas de salir adelante y por encontrarse siempre pendiente de mí, con su cariño y comprensión en la etapa final de la carrera.

A la Escuela Politécnica Nacional y todos los profesores quienes supieron hacerme llegar no solo conocimiento sino también valores para la vida.

De manera especial agradezco a mi director Dr. Diego Reinoso por toda la paciencia, apoyo y colaboración desde el principio hasta la culminación de este proyecto. Y además por todos los consejos brindados a lo largo de la carrera.

Danny Simbaña

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN.....	VI
ABSTRACT.....	VII
1. INTRODUCCIÓN.....	1
1.1 OBJETIVO GENERAL	2
1.2 OBJETIVOS ESPECÍFICOS	2
1.3 ALCANCE	2
1.4 MARCO TEÓRICO	3
1.4.1 UAV - VEHÍCULOS AÉREOS NO TRIPULADOS	3
1.4.2 SISTEMA DE COMUNICACIÓN.....	6
1.4.3 PRESUPUESTO DEL ENLACE.....	8
2. METODOLOGÍA.....	12
2.1 REQUERIMIENTOS	12
2.2 DISEÑO DE LA INTERFAZ.....	13
2.3 IMPLEMENTACIÓN DE LA APLICACIÓN	19
2.3.1 FUNCIONES.....	19
2.3.2 IMPLEMENTACIÓN DE LAS INTERFACES.....	23
3. RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	34
3.1 RESULTADOS.....	34
3.1.1 PRUEBAS DE FUNCIONAMIENTO	34
3.1.2 ANÁLISIS DE ARCHIVOS Y GRÁFICAS	41
3.1.3 ANÁLISIS DE PRUEBAS CON EL MISMO RECORRIDO.....	47
3.2 CONCLUSIONES	52
3.3 RECOMENDACIONES	53
4. REFERENCIAS BIBLIOGRÁFICAS.....	54
5. ANEXOS	56

RESUMEN

El presente proyecto incluye el diseño e implementación de una interfaz en Matlab que permite el procesamiento de archivos telemétricos producidos por planes de vuelo de vehículos aéreos no tripulados (UAV), enfocándose en la proyección de gráficas de los parámetros que intervienen en la calidad del enlace de telemetría.

En el capítulo 1 se encuentra información teórica general acerca de la definición, clasificación y aplicaciones de los UAV. Además, se presenta información de sus sistemas de comunicación y conceptos acerca del presupuesto del enlace en donde intervienen diferentes aspectos que influyen en las pérdidas en el enlace de comunicación.

En el capítulo 2 se describe el proceso de diseño de interfaz en donde se presentan las diferentes funcionalidades que intervienen dentro de la visualización de las gráficas. Así mismo se incluye el análisis y selección de los parámetros más importantes sobre la calidad del enlace de telemetría, para su posterior implementación mediante la ayuda del App Designer en Matlab.

En el capítulo 3 se presenta las pruebas de funcionamiento de la interfaz y análisis de resultados enfocado a la identificación de correlaciones entre los diferentes parámetros. Finalmente, se hace uso de la aplicación con un archivo log reciente, a fin de determinar cual es la mejor combinación de alineación en las antenas presentes en el UAV y en la estación base.

PALABRAS CLAVE: UAV, Log, Telemetría, Calidad de enlace, RSSI.

ABSTRACT

This project includes the design and implementation of an interface in Matlab that allows the processing of telemetric files produced by flight plans of unmanned aerial vehicles (UAV), focusing on the projection of graphs of the parameters involved in the quality of the telemetry link.

Chapter 1 contains general theoretical information about the definition, classification and applications of UAVs, as well as information about their communication systems and concepts about the link budget, where different aspects that influence losses in the communication link are involved.

Chapter 2 describes the interface design process where the different functionalities involved in the visualization of the graphics are presented, as well as the analysis and selection of the most important parameters about the telemetry link, for later the implementation with the help of the App Designer in Matlab.

Chapter 3 presents the functional tests of the interface and the analysis of results focused on the identification of correlations between the different parameters. Finally, the application is used with a recent log file, in order to determine the best combination of antenna alignments present in the UAV and in the base station.

KEYWORDS: UAV, Log, Telemetry, Link Quality, RSSI.

1. INTRODUCCIÓN

Los vehículos aéreos no tripulados (UAV), comúnmente conocidos como drones, se han vuelto cada vez más importantes en una amplia gama de industrias y aplicaciones en los últimos años. La versatilidad y eficiencia de los drones los hacen ideales para tareas que podrían ser peligrosas, difíciles o imposibles de realizar para los humanos: desde la entrega de paquetes a áreas remotas, inspección de líneas eléctricas y plataformas petroleras, hasta la captura de imágenes aéreas para películas y fotografías. Los drones han demostrado ser herramientas valiosas para muchas empresas y organizaciones. Hoy en día, los UAV se utilizan cada vez más para la seguridad pública y la respuesta a emergencias, proporcionando evaluaciones aéreas rápidas después de desastres naturales y otras emergencias. Los beneficios de la tecnología UAV son numerosos y crecientes, lo que la convierte en una herramienta indispensable en la actualidad.

Ahora bien, este proyecto pretende centrarse en el sistema de comunicación de un UAV, específicamente en los enlaces de telemetría, los cuales juegan un papel fundamental dentro de los UAVs. Estos enlaces permiten la transmisión remota de datos e información entre el dron y su operador o entre el dron y una radio base donde se encuentra todo el sistema de control en tierra. Los datos de telemetría proporcionan información en tiempo real sobre el estado y el rendimiento del UAV, incluida su posición, altitud, velocidad y niveles de potencia, entre otros parámetros que se guardan en archivos de datos denominados "logs", los cuales vienen entregados en varios formatos. Ahora bien, esta información es esencial para garantizar una operación segura y eficiente, lo que permite a los operadores tomar decisiones informadas, solucionar problemas y realizar los ajustes necesarios en los planes de vuelo o en operaciones en tiempo real. No se puede subestimar la importancia de los enlaces de telemetría, ya que forman la columna vertebral de una misión de dron exitosa y proporcionan el vínculo crítico entre el dron y su operador.

Es por ello que, en este proyecto se pretende crear una interfaz de visualización de estos datos telemétricos que interviene específicamente en la calidad del enlace de comunicación, de tal manera que se seleccione los parámetros más importantes en base a un análisis previo y se muestren al usuario, con la posibilidad de la edición de intervalos de tiempo o diferentes valores de frecuencia, con el fin de analizar posibles correlaciones entre varios parámetros de un mismo archivo o comparar estos mismos parámetros entre archivos.

1.1 OBJETIVO GENERAL

Programar en Matlab una interfaz gráfica para el procesamiento de la información de la calidad del enlace de telemetría de un UAV.

1.2 OBJETIVOS ESPECÍFICOS

1. Diseñar la interfaz para el procesamiento de los logs telemétricos en Matlab.
2. Implementar el código de la interfaz de Matlab.
3. Analizar los resultados de las pruebas realizadas.

1.3 ALCANCE

En este componente se codifica una interfaz gráfica en Matlab para procesamiento de la información telemétrica denominados "logs", a fin de evaluar la calidad del enlace en función del posicionamiento físico de la antena en el UAV. Los datos son proporcionados por la controladora de vuelo, la cual tiene instalada el firmware de ardupilot. Es importante mencionar que la información será entregada al estudiante de tal manera que dentro de este proyecto no se considera la implementación del dron ni la antena. El desarrollo de este componente se trabaja con 4 fases:

Fase de diseño

En esta fase se determinará los parámetros que influyen principalmente en la calidad del enlace para posteriormente realizar un bosquejo de la interfaz gráfica en la que se visualice dicha información.

Fase de implementación

En esta fase se realiza la codificación en Matlab de tal manera que se incluyan las funciones de selección del intervalo de tiempo de la información y visualización de gráficas de dos logs telemétricos diferentes.

Fase de prueba

En esta fase se considera la información telemétrica de dos misiones de vuelo autónomas con el mismo recorrido, pero con la antena colocada en dos posiciones diferentes dentro del cuerpo físico del UAV. Para posteriormente hacer uso de la interfaz gráfica diseñada e implementada.

Fase de análisis de resultados

En esta fase, se evaluará las gráficas resultantes a fin de determinar cuál posición de la antena produce una mejor calidad de enlace. Finalmente se agregará conclusiones y recomendaciones.

1.4 MARCO TEÓRICO

1.4.1 UAV - VEHÍCULOS AÉREOS NO TRIPULADOS

Un vehículo aéreo no tripulado es un dispositivo semi-inteligente que se puede controlar de forma remota. En general se trata de todo tipo de vehículo, independientemente de su tamaño, que no integre pasajeros ni pilotos de tal manera que pueda desplazarse de forma autónoma o ser controlado remotamente desde tierra [1].

Dentro de las especificaciones básicas para la construcción física de un dron se tiene los siguientes elementos

1. El chasis: Es el esqueleto del dron y puede tener diferentes formas y características. En particular, los marcos se distinguen por su número de brazos. El cual puede estar hecho de madera, plástico, aluminio o incluso fibra de carbono.
2. El sistema de propulsión: Está compuesto principalmente por:
 - Motores: es el elemento que a través de la conversión de energía eléctrica en energía mecánica produce el movimiento de las hélices para la puesta en vuelo del UAV e influyen directamente en la carga máxima soportada y en el tiempo de en el aire.
 - Hélices: son las encargadas de producir el empuje.
 - Controladores electrónicos: son encargados de gestionar la velocidad, dirección y giro de los motores.
 - Batería: suministra la energía necesaria para los demás componentes eléctricos.
3. El controlador de vuelo: Está compuesto por un circuito integrado con un microprocesador, sensores y pines de entrada y salida. Para configurar un dron, se debe usar un software para transferir los datos a la placa del controlador [2].

1.4.1.1 Clasificación

Un UAV se puede clasificar en base al tipo de ala que se está empleando o al tipo de control que integra el vehículo [3].

1.4.1.1.1 Tipo de ala

- Ala fija: En esencia el principio de funcionamiento se basa en la sustentación de un avión en la que las alas no tienen ningún tipo de movimiento y están colocadas a los extremos laterales del dron. Para que se produzca el movimiento del equipo se incorpora alerones y servomotores; se incluye una hélice la cual se encuentra posicionada de forma perpendicular al suelo. Es importante mencionar que este tipo de UAV debe desplazarse a una mayor altura y gran velocidad para mantener el ritmo de vuelo lo cual representaría una desventaja al momento del uso de una cámara para fotos o vídeos.
- Multirroto: Este tipo de UAV emplean, varios motores para su puesta en vuelo y movimiento, generalmente las hélices están distribuidas a los extremos del equipo de manera paralela al suelo. Un equipo muy común es el denominado cuadricóptero el cual integra dos hélices que vuelan en sentido horario y dos de hélices que vuelan en sentido antihorario lo que le proporciona el empuje para proceder a la elevación y al desplazamiento. Es importante mencionar que se tiene como ventaja que estos equipos se pueden mantener elevados en una posición fija en el aire y que no es necesario de gran altura para su puesta en vuelo, por el contrario, se tiene como desventaja que el emplear varios motores reduce drásticamente la autonomía del UAV [4].

1.4.1.1.2 Tipo de control

- Manual: Se basa en el manejo de controles roll, pitch, yaw y throttle los cuales representan diferentes posiciones de las hélices para lograr el desplazamiento del equipo. Para ello es importante el uso de un control desde tierra que da instrucciones al UAV mediante un enlace de radio frecuencia. Además, este debe estar siendo operado por un piloto el cual debe estar capacitado teniendo en cuenta que el dron no se aleje de los límites de cobertura del enlace de comunicación de radiofrecuencia. Es importante mencionar que en este tipo de modo es indispensable el contacto visual operario-UAV. En general, este tipo de aeronaves se los comercializa

para fines de entretenimiento por lo que suelen ser bastante fáciles de conseguir y a bajo costo.

- Automático: En este modo de control se emplea un plan de vuelo previamente cargado en el equipo en el cual se especifica la trayectoria de desplazamiento y el posicionamiento en diferentes puntos de interés. Esta información debe ser almacenada por la controladora de vuelo, la cual integra un GPS que intervendrá en la navegación. Aunque en este modo no es importante que se tenga contacto visual operario-UAV si es necesario que se encuentre un piloto pendiente en tierra que pueda intervenir en el control en caso de cualquier evento no considerado o asimismo en la modificación del plan de vuelo de forma remota.
- Autónomo: Este modo de control es parecido al modo automático debido a que de igual forma se integra un plan de vuelo, pero se diferencia en que estos UAVs llevan una controladora de vuelo más inteligente que integra gran variedad de sensores que le permitan al equipo de manera completamente autónoma la configuración del plan de vuelo en base a las condiciones climáticas, terrenas, etc., sin ninguna intervención humana [1].

1.4.1.2 Aplicaciones

Inicialmente la creación de estos equipos aéreos fue enfocada en el ámbito militar con el objetivo de realizar reconocimientos de terreno en áreas de poco acceso e inclusive integrando armamento para el combate aéreo. Hoy en día el enfoque de aplicación de un UAV es bastante variado debido principalmente a su gran potencial de desplazamiento de una manera rápida y eficaz sobre cualquier tipo de terreno y además resaltando la característica de un costo de operación considerablemente bajo en comparación a cualquier tipo de sistema tripulado [5]. En muchos casos se ofrece servicios de video que proporcionan datos visuales desde una toma aérea en tiempo real, por otro lado es muy común también que se integren sensores dentro del mismo a fin de recoger datos que pueden servir para el estudio en diversos enfoques de trabajo, de tal manera que estas características han sido aprovechadas ya no únicamente bajo intereses militares sino también civiles como: servicios profesionales, investigaciones científicas, usos recreativos, telecomunicaciones, cuidado ambiental, soporte médico y rescate [6].

1.4.2 SISTEMA DE COMUNICACIÓN

Dentro del entorno del UAV, este sistema se refiere a todos los elementos que intervienen para realizar uno o varios enlaces de comunicación, entre la estación terrena y el equipo aéreo. Para ello se emplea el uso de las señales de radiofrecuencia el cual actúa como medio de transmisión para los diferentes enlaces relacionados a una función específica como, por ejemplo: el control manual del equipo, telemetría, video, etc. Es importante mencionar que cada uno de estos enlaces de comunicación están integradas dentro de diferentes frecuencias de transmisión y asignaciones de ancho de banda.

Existen dos tipos de enlace. De tipo Uplink que se encarga de enviar la información como por ejemplo las señales de control del equipo desde la estación terrena hasta el UAV. Por otro lado, están los Downlink los cuales se encargan de devolver la información como, por ejemplo: datos de sensores integrados en el UAV, captura de imágenes, video, etc [7].

1.4.2.1 Enlace de Telemetría

La telemetría es un proceso que se realiza de forma automática, el cual es utilizado para recopilación de mediciones de sensores de forma remota. Para este caso los datos se envían entre el UAV y la estación terrena, para ser supervisados y analizados con el fin de mejorar el rendimiento de vuelo del UAV. Este enlace de comunicación generalmente es de frecuencias bajas debido a que permite una propagación de señal a más distancia, pero con la desventaja que se pueden incluir únicamente una cantidad limitada de datos a una baja tasa de velocidad, aunque es suficiente para el propósito de monitorear el estado del UAV. Cabe mencionar que la elección de una banda de frecuencia debe ir dentro de los lineamientos establecidos por las entidades regulatorias de cada país. En el caso del Ecuador son la ARCOTEL a través del Plan Nacional de Frecuencias (PNF). En el cual se especifica las bandas para operación libre que no interfieren con actividades de terceros, tal y como se puede observar en la Tabla 1.1 [8]

Tabla 1.1 Bandas libres del Plan Nacional de Frecuencias

Banda	Unidades
430 - 440	MHz
915 - 928	MHz
1240 - 1300	MHz
2300 - 2450	MHz
5650 - 5925	MHz
10 - 10.5	GHz
24 - 24.25	GHz

Para el funcionamiento del enlace de telemetría se recomienda escoger bandas de frecuencia que son utilizadas también por los equipos relacionados con los UAV, tal y como se presenta en la tabla 1.2, con el objetivo de seguir en línea con el desarrollo y la proyección del uso de esta tecnología para aumentar el nivel comercial de los componentes de tal manera que se vaya reduciendo sus costos de fabricación y se mantenga la accesibilidad a dichos equipos [3].

Tabla 1.2 Bandas de frecuencia recomendadas para el enlace de telemetría

Banda	Uso
430 – 440 [MHz]	Enlaces de telemetría UAV
915 – 928 [MHz]	Enlaces de video y telemetría de UAV
2.3 – 2.45 [GHz]	Enlaces de video y telemetría de UAV con WiFi o Bluetooth

1.4.2.2 Enlace de video

Los equipos UAV dan posibilidad a un sin fin de aplicaciones enfocadas al sector audiovisual con el objetivo de capturar imágenes desde ángulos imposibles en ubicaciones geográficas de difícil acceso.

Tabla 1.3 Bandas de frecuencia recomendadas para el enlace de video

Banda	Uso
915 – 928 [MHz]	Enlaces de video y telemetría de UAV
1.24 – 1.3 [GHz]	Enlaces de video de UAV GPS
2.3 – 2.45 [GHz]	Enlaces de video y telemetría de UAV con WiFi o Bluetooth
5.65 – 5.92 [GHz]	Enlaces de video de UAV Redes 5G

Este enlace de comunicación también se encuentra ligado a un grupo de bandas de frecuencias no licenciadas que cumplen características específicas para la aplicación de videovigilancia, tal y como se presenta en la tabla 1.3. Se requiere de una alta frecuencia debido a la alta tasa de datos que se debe transmitir, en general se considera a los 5.8 GHz como una frecuencia adecuada en un enlace de video en un UAV debido a que es apta para operar en sectores de vegetación considerable además de que se puede implementar en dispositivos RF que no ocupan gran tamaño físico dentro del UAV y se le puede utilizar dentro de un rango de distancia moderada [3].

1.4.3 PRESUPUESTO DEL ENLACE

Un presupuesto de enlace es una descripción de todas las ganancias y pérdidas de un transmisor durante su estado activo es decir mientras se comunica con un receptor a través de un medio de transmisión, es por ello que para su cálculo se debe tener en cuenta todas las ganancias y pérdidas de potencias desde el punto de transmisión hasta el punto de recepción. En el enlace entre una estación terrena y un UAV se considera las pérdidas producidas por el medio transmisor, es decir el aire y las diferentes atenuaciones causadas por factores dentro del mismo.

Potencia recibida (dBm) = Potencia transmitida (dBm) + ganancias (dB) - pérdidas (dB)

El presupuesto del enlace es un parámetro imprescindible para determinar la sensibilidad de los elementos que intervienen en los enlaces de tal manera que se consiga una comunicación eficiente entre el transmisor y el receptor [9].

Ahora bien, si se enfoca este concepto para el enlace de comunicación en un sistema con un UAV se muestra la fórmula de la potencia en lado del receptor en la ecuación 1.1 [10].

$$P_R = G_T + P_T - L_T - L_F - L_R - L_A - L_O + G_R \quad (1.1)$$

Donde: P_R es la potencia en el receptor y P_T la potencia en el transmisor. Se nota que se incluyen las ganancias en ambos extremos del enlace donde: G_R y G_T representan a la ganancia de las antenas en el receptor y transmisor respectivamente [10].

Se pone especial énfasis en las pérdidas, debido a que son las que incurren en atenuaciones dentro del presupuesto del enlace y principalmente porque aparecen de manera inintencional al diseño de los componentes que componen el enlace, afectando directamente la comunicación.

- L_F : Se refiere a las pérdidas en el espacio libre en un enlace con línea de vista.
- L_R : Se refiere a las pérdidas causadas por la lluvia.
- L_A : Se refiere a las pérdidas causadas por la atmósfera gaseosa principalmente vapores de agua o aire seco.
- L_O : Se refiere a la pérdida por desvanecimiento, la cual puede ser a gran escala donde principalmente afecta el factor de la distancia y los elementos que interviene dentro de la línea de vista como edificios y colinas. También puede ser a pequeña escala donde aparecen los fenómenos de multitrayecto, reflexiones y dispersiones de la señal.
- L_T : Se refiere a las pérdidas totales tanto en el enlace ascendente como en el enlace descendente que engloba diferentes eventos como: el desajuste de la polarización de las antenas, pérdida de línea de vista, pérdidas en los cables y conexiones que se integran en los circuitos de los equipos, pérdida por el alimentador del receptor.

1.4.3.1 Pérdidas en espacio libre

La denominada pérdida en el espacio libre es al desvanecimiento de la intensidad de una señal en la recepción a medida que se aumenta la trayectoria entre la antena receptora y transmisora. Teniendo en cuenta que la onda electromagnética va viajando en línea recta [10].

Usualmente también es conocida como pérdida de dispersión y se mide en decibelios siguiendo la ecuación 1.2.

$$L_f(dB) = 92.45 + 20 \log f(\text{GHz}) + 20 \log d(\text{Km}) \quad (1.2)$$

Donde:

- L_f : Pérdidas en el espacio libre en decibelios.
- f : Frecuencia en GHz.
- d : distancia entre transmisor y receptor en Km.

1.4.3.2 Clasificación de antenas

Una antena es un dispositivo que se encuentra en el cuerpo físico en la etapa de transmisión y recepción en un sistema de comunicación y actúa radiando energía de tal manera que transmite ondas electromagnéticas a partir de señales eléctricas o viceversa.

Se puede clasificar las antenas en base a la forma en la que está construida, tamaño y forma [11]. A continuación, en la tabla 1.4 se presentan varios ejemplos de antenas.

Tabla 1.4 Tipos de Antenas

Tipo de antena	Ejemplos
Antenas de alambre	Antena monopolo y dipolo, antena helicoidal.
Antenas de apertura	Guías de onda, antena horn
Antenas reflectoras	Reflector parabólico
Antenas de lentes	Lentes de plano convexo y cóncavo, convexo – convexo, cóncavo – cóncavo
Antenas de micro tira	Antena patch de forma circular o rectangular

A partir de aquí comienza la selección de la antena acorde a las aplicaciones requeridas teniendo en cuenta las especificaciones de tamaño, potencia, frecuencia, etc. A continuación, se presentan varios ejemplos de antenas de uso comercial en el sector de los UAV enfocado a enlaces de telemetría.

- Antena monopolo MP900

Es una antena omnidireccional, trabaja en el rango de 902 – 928 MHz con una ganancia de 3 dBi. Viene integrada con tecnología de multi polarización de antena para mantener un rendimiento continuo e ideal para aplicaciones de comunicaciones en UAV sin necesidad de una línea de vista [12].

- Harris Aerial 900 MHz 3 dBi Base Antenna

Antena dipolo de media onda que trabaja en el rango de 902 - 928 MHz con una ganancia de 3 dBi, de tamaño pequeño de 15 cm y peso ligero de 28 g, ideal para integraciones en el cuerpo del UAV. Adicionalmente presenta características de reducción de ruido y polarización vertical [13].

- Antena monopolo 900 MHz 2.1 dBi RPSMA

Antena monopolo que trabaja en el rango de 902 - 928 MHz, con una ganancia de 2.1 dBi (RPSMA). Perfecto para drones UAV/FPV por su tamaño de 5 cm y bajo peso de 10 g. Se recomienda montarlo en el plano horizontal para asegurar un

mejor rendimiento, Así mismo presenta una polarización vertical y características de reducción de ruido [14].

1.4.3.3 Posicionamiento físico de las antenas

El correcto posicionamiento físico de la antena en el UAV garantiza que la señal sea recibida de manera óptima. Generalmente se debe procurar que una antena tipo dipolo se encuentre fija de manera perpendicular al cuerpo del drone en los dos tipos: ala fija y multirroto. Además, dicha posición es ideal para antenas con polarización lineal o circular. Y si se trata de una antena específica para el enlace de video, se recomienda colocarla encima de la cámara, esto asegura un rendimiento óptimo para aplicaciones de video vigilancia [15].

En el caso que se trate del enlace de telemetría se recomienda fijar la antena lo más alejada del marco posible de tal manera que la antena se encuentre lo más sobresalida que se pueda y que se trate de siempre trazar una línea recta entre la antena de transmisión en el UAV y la antena de recepción en la estación terrena. Se debe procurar que este enlace no se vea interpuesto por el cuerpo físico del drone [15].

2. METODOLOGÍA

En la presente sección del documento se detalla la metodología que se utilizó para el desarrollo del proyecto, que tiene como objetivo programar en Matlab una interfaz gráfica para el procesamiento de la información de la calidad del enlace de telemetría de un UAV.

De tal manera que se inicia determinando los principales parámetros que intervienen dentro de los datos telemétricos, es decir se va a seleccionar los datos más importantes que tengan que ver con la calidad del enlace dentro de los archivos de registro o también denominados archivos log, los cuales son los que contienen mensajes sobre el sistema o en este caso todos los factores que intervengan y produzcan datos medibles en el desarrollo de un plan de vuelo de un UAV, para de esta manera crear gráficas que permitan su visualización en función del tiempo.

Para ello se realizará un bosquejo de la interfaz gráfica en el que se considera la integración de ventanas, cajas de visualización y botones de activación, que inicialmente permita la carga del archivo log, para de manera seguida graficar los datos seleccionados e indicar información. Asimismo, es necesario mencionar que se realizará el bosquejo para la interfaz, que incluye la comparación de 2 archivos log diferentes.

Posteriormente, se va a proceder a la respectiva implementación de la aplicación mediante la ayuda del “App Designer” de Matlab el cual es una herramienta bastante didáctica e interactiva, que proporciona las herramientas necesarias tanto para la adecuación de componentes visuales dentro de la interfaz, así como la programación del comportamiento de la aplicación.

2.1 REQUERIMIENTOS

Dentro de los requerimientos para la implementación se considera el software Matlab en su versión 2020 B o posterior. Cabe mencionar que va a ser necesario el uso del “Mapping Toolbox” que proporciona algoritmos y funciones para el manejo de datos geográficos, lo cual va a ser imprescindible teniendo en cuenta que este tipo de información es de vital importancia dentro de los datos telemétricos.

Por parte de los requerimientos específicos para el diseño e integración de funcionalidades en la interfaz. Inicialmente, se debe tener la opción para la carga de un solo archivo o dos archivos en la aplicación, dicha elección debe ser dada por el usuario, teniendo en cuenta que el tipo de archivo de entrada debe estar en formato (.mat). De manera seguida, se requiere la proyección de gráficas de los datos telemétricos con el objetivo de poder realizar

una correlación, por lo que es imprescindible el uso de un mismo eje de tiempo, de preferencia en minutos. Asimismo, se requiere la posibilidad que el usuario pueda editar el intervalo de tiempo de la gráfica y además se requiere la visualización de información de valores importantes como: máximo, mínimo y promedio. Finalmente, se requiere de una segunda ventana que permita la selección de dos archivos, que integre las mismas funcionalidades mencionadas anteriormente.

2.2 DISEÑO DE LA INTERFAZ

Para el inicio del diseño de la interfaz es necesario identificar los parámetros que más influyen en la calidad del enlace, como se puede observar en la figura 2.1 se encuentra una muestra de los datos incluidos en un archivo log, los cuales fueron generados por la controladora a medida que se iba desarrollando el plan de vuelo.

Name	Value
ACC1_Label	6x1 cell
ACC2_Label	6x1 cell
ACC3_Label	6x1 cell
AETR	45509x7 double
AETR_Label	7x1 cell
AHR2_Label	12x1 cell
AOA	45509x4 double
AOA_Label	4x1 cell
ARM_Label	4x1 cell
ARSP	1820x8 double
ARSP_Label	8x1 cell
ATRP_Label	8x1 cell
ATT	45509x10 double
ATT_Label	10x1 cell
BAR2_Label	9x1 cell
BAR3_Label	9x1 cell
BARO	3640x9 double
BARO_Label	9x1 cell
BAT	1820x8 double
BAT2_Label	8x1 cell
BAT_Label	8x1 cell
BCL2_Label	13x1 cell
BCL_Label	13x1 cell
BCN_Label	11x1 cell
CAM	363x12 double
CAM_Label	12x1 cell
CMD	161x12 double
CMD_Label	12x1 cell
CTUN_Label	45509x10 double
CTUN_Label	10x1 cell
DMS_Label	16x1 cell
DSF	1788x9 double
DSF_Label	9x1 cell
MAG2_Label	13x1 cell
MAG3	1820x13 double
MAG3_Label	13x1 cell
MAG_Label	13x1 cell
MODE	6x5 double
MODE_Label	5x1 cell
MSG	171x3 double
MSG1	1x171 cell
MSG_Label	3x1 cell
NKFO_Label	15x1 cell
NKFI_Label	16x1 cell
NKJ2_Label	15x1 cell
NKJ3_Label	13x1 cell
NKJ4_Label	15x1 cell
NKJ5_Label	14x1 cell
NKJ6_Label	16x1 cell
NKJ7_Label	15x1 cell
NKJ8_Label	13x1 cell
NKJ9_Label	15x1 cell
NKQ1_Label	6x1 cell
NKQ2_Label	6x1 cell
NTUN	45509x9 double
NTUN_Label	9x1 cell
OF_Label	7x1 cell
ORGN	[1078,225771109,1,-0.4998,-78.215...
ORGN_Label	6x1 cell
PARM	788x2 cell
PIDA_Label	8x1 cell
PIDP_Label	45509x8 double
PIDP_Label	8x1 cell
PIDR_Label	45509x8 double
PIDR_Label	8x1 cell
PIDR_Label	8x1 cell
PIDR_Label	45509x8 double
PIDR_Label	8x1 cell
PIDR_Label	45509x8 double
PIQV_Label	8x1 cell
PM	364x8 double
PM_Label	8x1 cell
POS	45509x7 double
POS_Label	7x1 cell
POWR	1820x5 double
POWR_Label	5x1 cell
PRX_Label	14x1 cell
QTUN_Label	13x1 cell
RAD	1821x9 double
RAD_Label	9x1 cell
RALY_Label	7x1 cell
RATE_Label	14x1 cell
RCIN	45509x16 double
RCIN_Label	16x1 cell
RCOU	45508x16 double
RCOU_Label	16x1 cell
RFND_Label	6x1 cell
RPM_Label	4x1 cell
RSSL_Label	3x1 cell
SBFI_Label	14x1 cell
SBPH_Label	5x1 cell
SBRE_Label	7x1 cell
SBRM_Label	9x1 cell
SBRM_Label	16x1 cell
Seen	42x1 cell
SIM_Label	12x1 cell
SONR_Label	6x1 cell
SRTL_Label	9x1 cell
STAT	9106x10 double
STAT_Label	10x1 cell
STRT	[1076,225770998,5,81]
STRT_Label	4x1 cell

Figura 2.1. Datos dentro de un archivo Log

Se puede identificar gran cantidad de celdas que representan sub-archivos que contienen diferentes tipos de información, en este caso siguiendo la dirección del proyecto se ha podido destacar principalmente el parámetro “RAD” que se refiere a las estadísticas de radio de telemetría. Como se puede observar en la figura 2.2 se tiene 9 diferentes parámetros en el cual se destaca los literales:

- “TimeUS”: línea de tiempo en microsegundos desde que inicia hasta que termina la toma de datos de este sub-archivo, lo cual está directamente relacionado con la frecuencia de muestreo de la información. Cabe mencionar que cada sub-archivo tiene su propia línea de tiempo.

- “RSSI”: por sus siglas en inglés es el indicador de fuerza de la señal recibida, es decir es utilizado para medir el nivel de potencia, en el extremo de la comunicación en la que se está tomando los datos.
- “RemRSSI”: es el mismo indicador de fuerza de la señal recibida, pero en el extremo opuesto de la comunicación del lado que se está tomando los datos.
- “Noise”: el piso de ruido, qué hace referencia al nivel mínimo de potencia qué debe tener la señal para que exista comunicación, en el extremo del que se está tomando los datos.
- “RemNoise”: es el mismo piso de ruido, pero en el extremo opuesto de la comunicación del lado que se está tomando los datos.

	1	2	3	4
1 LineNo				
2 TimeUS				
3 RSSI				
4 RemRSSI				
5 TxBuf				
6 Noise				
7 RemNoise				
8 RxErrors				
9 Fixed				
10				
11				

Figura 2.2. Parámetros que contiene el sub-archivo “RAD”

Una vez identificado el nivel de potencia, es necesario tener en cuenta que los valores proporcionados directamente del archivo log, se encuentran en unidades con la propia escala del RSSI, que son valores enteros que van de 1 a 256 representando la intensidad de la señal de tal manera que, de inicio es necesario la conversión de este valor a unidades en dBm, por lo que se emplea la ecuación 2.1 para todos los datos del sub-archivo RAD, excepto el tiempo.

$$RSSI(dBm) = \left(\frac{RSSI}{1.9} \right) - 127 \quad (2.1)$$

Ahora bien, se considera que este parámetro está estrechamente relacionado con la distancia que existe entre el UAV y la radio base , de tal manera que para evidenciar esta

relación, es necesario la información de la distancia en el transcurso del tiempo. En este punto cabe mencionar que dentro de los archivos log, no existe un sub-archivo que proporcione la información de la distancia directamente, pero se puede realizar el cálculo en base a otros parámetros.

Inicialmente, se identifica un sub-archivo "GPS" el cual proporciona la información de latitud y longitud en coordenadas GPS en el transcurso del tiempo, de manera seguida se identifica el sub-archivo "BARO", el cual proporciona datos recopilados del barómetro y donde se puede encontrar la información de la altitud a la que está volando el UAV en el transcurso del tiempo.

Teniendo en cuenta lo antes mencionado se propone realizar el cálculo de la distancia en metros teniendo la información de latitud, longitud y altitud a la que se encuentra el UAV. De igual manera es necesario la misma información en el momento que se inicia la misión de vuelo, para ello se identifica el sub-archivo "ORGN" en el cual se encuentran los datos del origen de la navegación del vehículo. Una vez identificados los parámetros que intervienen del cálculo de la distancia, se propone el siguiente proceso:

1. Ubicar los valores de latitud y longitud en el origen y en el UAV en función del tiempo, obtener las distancias en línea recta entre estos dos puntos. Ubicar valores de altitud también en función del tiempo.
2. Aplicar el teorema de Pitágoras de tal manera que el cateto adyacente sea la distancia entre los puntos latitud-longitud y el cateto opuesto sea la altitud, para de esta manera calcular la hipotenusa, que vendría a ser la distancia entre el UAV y la estación terrena.
3. Obtener una distancia por cada dato de tiempo.

Por otro lado, también es importante las pérdidas en el espacio libre para un UAV, tal y como se detalla en la sección del presupuesto del enlace, en el que se encuentra que el cálculo depende de la frecuencia y la distancia entre el transmisor y el receptor. Ahora bien, teniendo en cuenta que se puede calcular la distancia y además se determina la banda de frecuencia de 902 - 928 MHz para la comunicación de datos telemétricos, se puede realizar el cálculo de las pérdidas en el transcurso del tiempo.

Otro parámetro importante dentro del enlace de radio comunicación es la polarización de las antenas. Específicamente en el enlace de telemetría, es muy común el uso de antenas con polarización lineal (vertical u horizontal), de tal manera que se requiere que en ambos extremos se tenga el mismo tipo de polarización, es decir ambas en forma vertical o ambas

en forma horizontal . Por lo que es imprescindible notar que debido al movimiento del vuelo del UAV, se puede perder fácilmente la alineación de las antenas, y en consecuencia el nivel de la señal recibida va a ir disminuyendo. En este caso, sería necesario el uso del sub-archivo “ATT” que contiene la información de rotación del movimiento del vehículo en 3 dimensiones, tal y como se puede observar en la figura 2.3.

En este caso, específicamente se utilizará el movimiento “roll” que permite el giro en el vuelo, y además el movimiento “pitch” que permite la elevación o descenso de la aeronave.

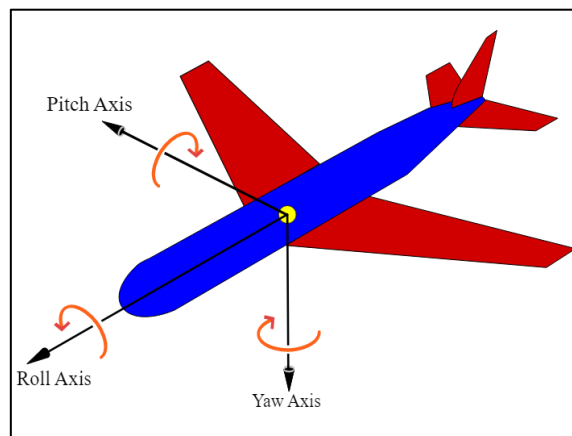


Figura 2.3. Ilustración de movimientos Roll, Pitch, Yaw

Una vez identificados los principales parámetros que influyen en la calidad del enlace, se presenta la propuesta de la lista de gráficas para la interfaz.

- Distancia entre UAV y estación base vs RSSI
- Pérdidas en el espacio libre vs RSSI
- ROLL vs RSSI
- PITCH vs RSSI
- RemNoise vs RemRSSI
- Noise vs RSSI

Teniendo en cuenta que en los primeros cuatro casos la información se encuentra en magnitudes completamente diferentes, pero que pueden guardar una estrecha relación, se propone graficarlos en el eje “y”, de tal manera que se tenga sus unidades de medida cada uno en un extremo de la gráfica. Siguiendo con las últimas dos gráficas no existe mayor dificultad debido a que ambos parámetros se encuentran en la misma unidad de medida. Por otro lado, en el eje “x” se propone siempre mantener la línea de tiempo en minutos.

A continuación, se presenta el bosquejo de la interfaz gráfica, incluyendo bloques de información y proyección de gráficas, botones de acción y su respectiva ubicación dentro del espacio. Para ello se propone la utilización de 3 ventanas de tal manera que se tenga inicialmente la presentación de la aplicación, tal y como se puede observar en la figura 2.4 en la que se detalla la información acerca del título, autor y tutor del proyecto. Cabe mencionar que la función principal de esta ventana es la selección por parte del usuario para la visualización de gráficas de un único archivo log o de 2 archivos log, de tal manera que dependiendo de la elección salte a una siguiente ventana.

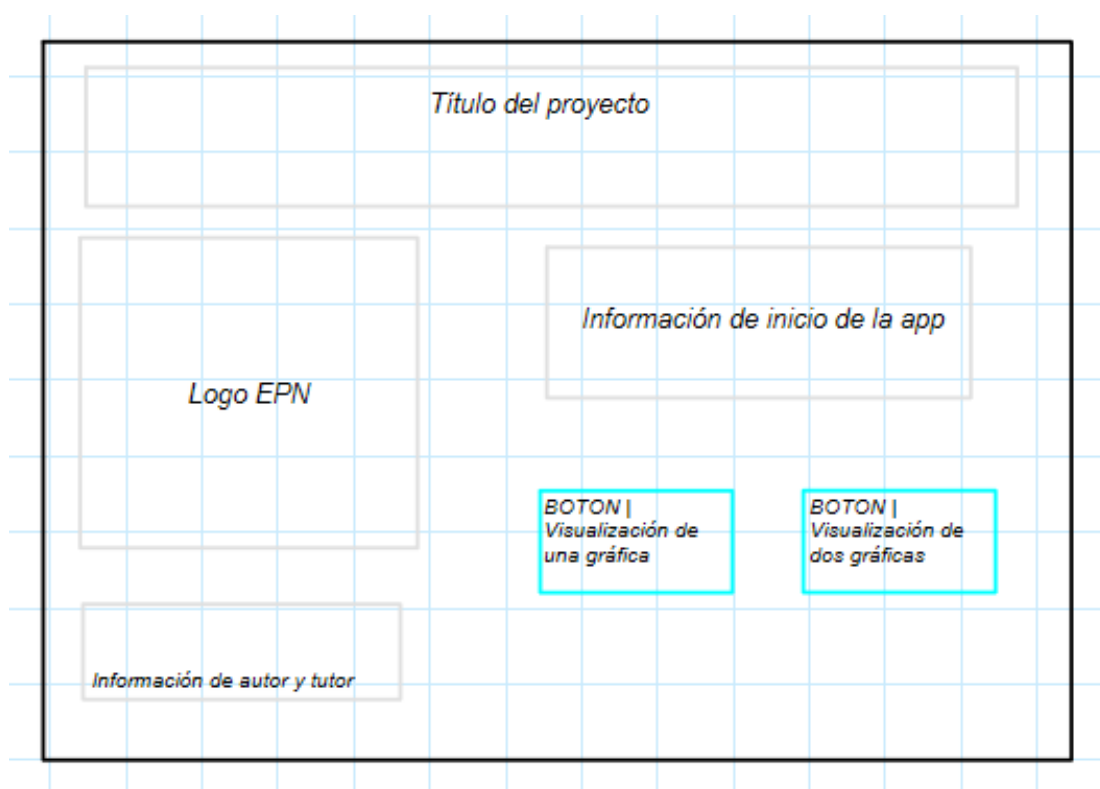


Figura 2.4. Bosquejo de la interfaz de presentación

El bosquejo de la interfaz para un único archivo log se presenta en la figura 2.5. en el que inicialmente se incluye un botón que permita la carga de un archivo log, de igual forma se presentan cajas de información como: el nombre del archivo cargado o el tiempo mínimo y máximo del que se tenga registro de información.

Además, se incluye botones con la lista de gráficas propuestas anteriormente, y espacios de configuración para la frecuencia y para la selección del intervalo de tiempo por parte del usuario. Finalmente, se incluye un espacio de visualización de las gráficas y una caja de información que contendrá los valores mínimos, máximos y promedios dependiendo del parámetro que se esté visualizando.

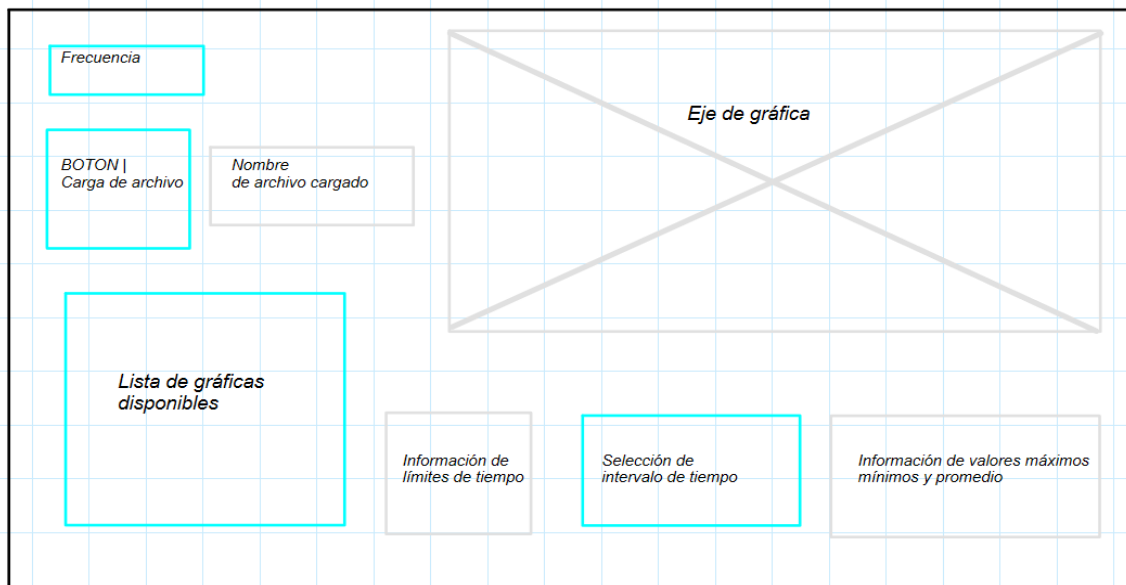


Figura 2.5. Bosquejo de la interfaz de visualización de gráficas de 1 archivo

En la figura 2.6 se presenta el bosquejo de la interfaz para la visualización de dos diferentes archivos log. Como se puede notar se duplica las cajas de información y botones que se tenía en la interfaz para un archivo. Aunque cabe mencionar que se mantiene una única lista de gráficas, esto con el objetivo de que se visualice los mismos parámetros para los dos archivos a la vez, de tal manera que se pueda realizar una comparación.

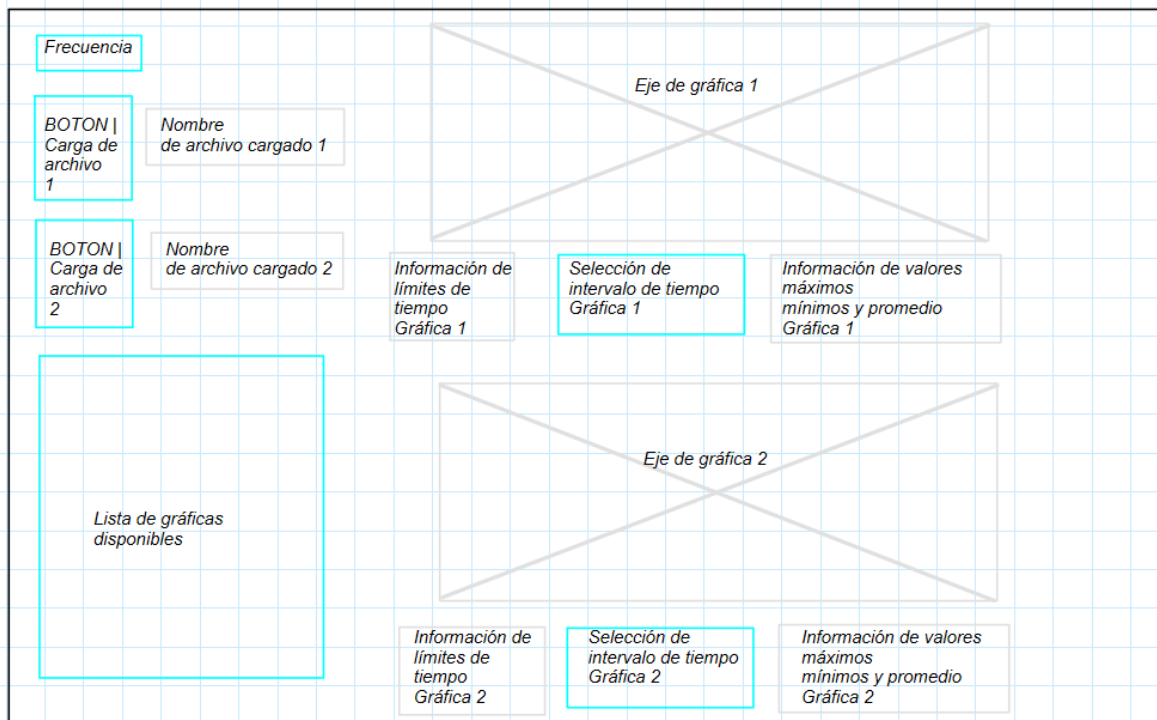


Figura 2.6. Bosquejo de la interfaz de visualización de gráficas de 2 archivos

2.3 IMPLEMENTACIÓN DE LA APLICACIÓN

Teniendo en cuenta lo que se requiere de la aplicación y los recursos que se tiene disponible para la implementación se procede a la codificación.

Para la creación de la lista de gráficas propuestas en la fase de diseño, hay que considerar que en cada gráfica se tiene dos datos de telemetría, por lo que se requieren vectores por separado que contengan estos datos, es decir un vector para RSSI, un vector para la distancia, un vector para las pérdidas, etc. Asimismo, es necesario un mismo vector de tiempo para poder graficar en el mismo plano y de manera sincronizada, los vectores mencionados anteriormente. Es por ello, que de manera general para una gráfica se requiere de tres vectores.

- vector de tiempo en el eje “x”.
- vector de datos telemétricos 1 en el eje “y”.
- vector de datos telemétricos 2 en el eje “y”.

Cabe mencionar que para realizar las gráficas en Matlab es necesario el uso de la función *plot*, la cual requiere que los vectores de datos de entrada del eje “y” y el vector de entrada del eje “x” tengan el mismo tamaño, es decir la misma cantidad de datos.

Las funciones y secciones de código necesarias para obtener dichos vectores se presentan a continuación.

2.3.1 FUNCIONES

Cómo se mencionó anteriormente en la fase de diseño, el principal problema de los sub-archivos es que cada uno contiene su propia línea de tiempo, es decir su propia frecuencia de muestreo. Por consiguiente, cada uno tiene un número de datos diferente.

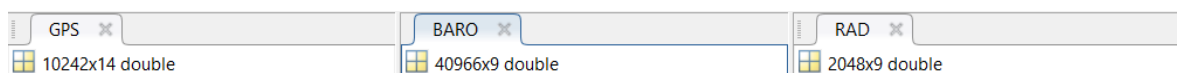


Figura 2.7. Ejemplo del tamaño de datos de los sub-archivos GPS, BARO, RAD

Como se puede observar en la figura 2.7, se muestra que tres de los principales sub-archivos, tienen un vector de datos distinto, por lo que es necesario realizar un ajuste de tal manera que en los vectores finales, tengan todos el mismo número de datos para que se pueda proceder a realizar la gráfica. En este punto cabe mencionar que no todos los logs son iguales de modo que es posible que en otro archivo se tengan tamaños de vectores diferentes al ejemplo de la figura 2.7.

Por otro lado, hay que notar que cada dato de tiempo está ligado a un dato de información telemétrica, y que de manera generalizada la columna número 2 de la matriz que representa a un sub-archivo, es la que contiene la información de los datos de tiempo.

Ahora bien, es fundamental notar que no necesariamente se tiene los mismos datos de tiempo como se puede observar en la figura 2.8. Aunque si se tiene valores muy parecidos, por ejemplo en la matriz RAD se tiene el tiempo 183864764 [us] y en la matriz GPS se tiene el tiempo 183884738 [us] donde ambos tiene una diferencia de 19974 [us] es decir 0.01974 [s] que para fines prácticos se lo considera irrelevante. Por lo que no existirá mayor error si se relacionan el dato de un sub-archivo con el tiempo más parecido de otro sub-archivo.

LineNo	1	2	3	4	5	6	7
1	1475	183864764	183	182	100	96	84
2	2072	184904647	182	182	91	101	83
3	2688	185964660	182	182	88	101	80
4	3218	186904579	183	183	98	102	83
5	3788	187885054	184	184	95	99	81

LineNo	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	1006	183085128		4	492899400	2125	19	0.570						
3	1147	183284614		4	492899600	2125	19	0.570						
4	1262	183484670		4	492899800	2125	19	0.570						
5	1379	183704744		4	492900000	2125	19	0.570						
6	1492	183884738		4	492900200	2125	19	0.570						

Figura 2.8. Datos de tiempo de los sub-archivos RAD y GPS

Teniendo en cuenta lo mencionado anteriormente se ve la necesidad de programar una función que ajuste y relacione los datos de tiempo de un sub-archivo con otro sub-archivo independientemente del número de datos.

Segmento de código 2.1. Función *igualtiempos*

```
function [A] = igualtiempos(C,C2)
% Verificar que los datos de entrada sean vectores con datos de TIEMPO
%C = Vector de entrada de tiempos fijos
%C2 = Vector con el que se compara para ir igualando tiempos en base a los tiempos del
vector C

for i=1:length(C)
    xx=C(i);% se toma uno por uno el tiempo del vector de entrada de tiempos fijos
    [minimo,position]=min(abs(C2-xx)); % se captura el valor mas cercano
    A(i)= position; %Se guarda en el vector de salida la posicion del valor mas
    cercano
end

end
```

Se presenta la función *igualtiempos* en el segmento de código 2.1. Su algoritmo de funcionamiento procesa dos vectores:

- C: que se refiere al *tiempo general*, es decir el vector de tiempo que va a ir fijo en todas las gráficas, el cual puede ser cualquiera de los vectores de tiempo de cualquiera de las matrices.
- C2: que se refiere al vector de tiempo de cualquier otra matriz, que se desea igualar al vector de tiempo general.

Esta función opera de tal modo que se va tomando uno por uno los datos del vector *tiempo_general* y encontrando el dato más parecido en el vector C2, para posteriormente ir guardando la ubicación de dicho dato. Como resultado se obtiene un vector de salida denominado A que tiene el mismo tamaño del vector tiempo general y contiene la posición de los datos de tiempo más parecidos de forma ordenada.

Bastaría con obtener las posiciones ya que estas vienen ligadas a toda la información telemétrica en dichos puntos, tal y como se observó en la figura 2.8. No obstante, se va a requerir de otro algoritmo que permita generar vectores de datos telemétricos en función de las posiciones obtenidas, para ello se presenta la función denominada *selecposicion* en el segmento de código 2.2.

Segmento de código 2.2. Función *selecposicion*

```
function [vec_igualado] = selecposicion(posiciones,datos)
%Permite la obtencion de data en base a las posiciones determinadas
for i=1: length(posiciones)
    vec_igualado(i)= datos(posiciones(i)); %guarda en un vector salida la data
end
end
```

Su algoritmo de funcionamiento procesa dos vectores:

- posiciones: se refiere al vector de salida de la función *igualtiempos*.
- datos: se refiere al vector de datos telemétricos contenido en el sub-archivo.

El algoritmo funciona de tal modo que se va accediendo a los datos telemétricos en una posición determinada a la vez y dicha información se va guardando en un vector de salida denominado *vec_igualado*. El resultado es un vector con datos de telemetría de un sub-archivo sincronizado con el tiempo general.

Finalmente, obteniendo los 3 vectores mencionados al inicio de la sección 2.3, únicamente haría falta una función que permita graficar dichos datos de manera sencilla. Para ello se

emplea la siguiente función denominada *graficar*, tal y como se presenta en el segmento de código 2.3, la cual tiene 12 parámetros como datos de entrada:

- app: Se refiere al nombre de axis donde se va a visualizar la gráfica.
- app1: Se refiere a la caja de información del dato máximo en la gráfica.
- app2: Se refiere a la caja de información de dato mínimo en la gráfica.
- app3: Se refiere a la caja de información de dato promedio en la gráfica.
- t: Vector de tiempo general.
- data1: Vector de datos de telemetría 1.
- data2: Vector de datos de telemetría 2.
- etq_y1: Nombre de los datos de telemetría 1.
- uni_y1: Nombre de la unidad de medida de los datos de telemetría 1.
- etq_y2: Nombre de los datos de telemetría 2.
- uni_y2: Nombre de la unidad de medida de los datos de telemetría 2.
- titulo: Título de la gráfica.

Segmento de código 2.3. Función *graficar*

```
function results =
graficar(app, app1, app2, app3, t, data1, data2, etq_y1, uni_y1, etq_y2, uni_y2, titulo)

tiempo_general=t; % vector de tiempo general
Fntlabel=15; % tamaño de etiqueta
Fnttitle=20; % tamaño del titulo

cla(app, 'reset'); % Reset del campo axis
yyaxis(app, 'left') %ajuste de eje y a la izquierda
plot(app, tiempo_general*1.6667e-8, data1) % grafica 1
ylabel(app, etq_y1+uni_y1, "FontSize", Fntlabel) % etiqueta eje y de grafica 1
yyaxis(app, 'right') % ajuste de eje y a la derecha
plot(app, tiempo_general*1.6667e-8, data2) % grafica 2
ylabel(app, etq_y2+uni_y2, "FontSize", Fntlabel) % etiqueta eje y de grafica 2
xlabel(app, 'Tiempo[min]', "FontSize", Fntlabel) % etiqueta eje x
grid(app, 'minor')
title(app, titulo, "FontSize", Fnttitle) % titulo
legend(app, etq_y1, etq_y2) %leyendas de las lineas
% Muestra de valores maximos minimos y promedios
app1.Value= etq_y1 + ": " + num2str(max(data1)) + " " + uni_y1 + " | " + etq_y2 + ": " +
num2str(max(data2)) + " " + uni_y2;

app2.Value = etq_y1 + ": " + num2str(min(data1)) + " " + uni_y1 + " | " + etq_y2 + ": " +
+ num2str(min(data2)) + " " + uni_y2;

app3.Value = etq_y1 + ": " + num2str(mean(data1)) + " " + uni_y1 + " | " + etq_y2 + ": " +
+ num2str(mean(data2)) + " " + uni_y2;

end
```

Tal y como se explicó en la sección de la lista de gráficas, para determinar la relación entre dos datos telemétricos de unidades de medida diferentes, se ve la necesidad de incluir dos líneas de datos dentro del mismo axis, para lo cual se ha empleado un ajuste al eje “y” a la izquierda o a la derecha según corresponda. Así mismo se va acomodando los nombres, etiquetas, leyenda y título dentro de la gráfica resultante, para finalmente obtener el valor máximo, mínimo y promedio e ir concatenando la información para que se pueda presentar de manera clara.

Ahora bien, es necesario mencionar que la anterior función únicamente servirá para mostrar las cuatro primeras gráficas, debido a que se tienen unidades distintas. A diferencia de las dos últimas que presentan datos en las mismas unidades por lo que se modifica un poco la función.

Segmento de código 2.4. Función *graficar2*

```
function results =
graficar2(app,app1,app2,app3,t,data1,data2,etq_y1,etq_y2,uni_y1,titulo)

tiempo_general=t; % vector de tiempo general
Fntlabel=15; % tamaño de etiqueta
Fnttitle=20; % tamaño del titulo

cla(app,'reset'); % Reset del campo axis
hold(app,'on');

    plot(app,tiempo_general*1.6667e-8,data1) % grafica 1
    plot(app,tiempo_general*1.6667e-8,data2) % grafica 2
    title(app,titulo,"FontSize",Fnttitle)% titulo
    legend(app,etq_y1,etq_y2) %leyendas de las lineas
    ylabel(app,etq_y1 + " ; " + etq_y2 + " " +
uni_y1,"FontSize",Fntlabel)
    xlabel(app,'Tiempo[min]',"FontSize",Fntlabel)% etiqueta eje x
    grid(app,'minor')

hold(app,'off');

% Muestra de valores maximos minimos y promedios
app1.Value= etq_y1 + ": " + num2str(max(data1)) + " " + uni_y1 mini | " + etq_y2 +": " +
num2str(max(data2)) + " " + uni_y1;
app2.Value = etq_y1 + ": " + num2str(min(data1)) + " " + uni_y1 + " | " + etq_y2 +": " +
+ num2str(min(data2)) + " " + uni_y1;
app3.Value = etq_y1 + ": " + num2str(mean(data1)) + " " + uni_y1 + " | " + etq_y2 +": " +
+ num2str(mean(data2)) + " " + uni_y1;

end
```

Como se puede observar en el segmento de código 2.4, solo es necesario la inclusión de una unidad de medida y ya no es necesario un ajuste al eje “y”. Únicamente se muestran las gráficas en el mismo axis.

2.3.2 IMPLEMENTACIÓN DE LAS INTERFACES

Inicialmente, se requiere programar la interfaz de presentación para ello se hace uso del app designer, mediante la herramienta *Label* para la introducción de textos. Tal y como se

puede observar en la figura 2.9 se presenta la configuración del contenido, tipo y tamaño de letra, color, etc; mediante el apartado de edición *Inspector*, el cual permite la configuración de todos los mensajes de texto incluidos en la ventana de una manera práctica y sencilla.



Figura 2.9. Configuración del título en la ventana de presentación

A continuación, en el segmento de código 2.5 se presenta la programación de los botones incluidos en la ventana de presentación. Como se observa en cada uno de los botones se llama a abrir una nueva aplicación denominada *appgeneral1* y *appgeneral2* que son las interfaces que contiene la visualización de gráficas para un archivo y dos archivos respectivamente.

Notar que en cada botón se utiliza la línea *delete(app)* que permite cerrar la aplicación actual, es decir va a cerrar la ventana de presentación cuando se elija cualquiera de las dos opciones. Esto con el objetivo de dejar únicamente una sola pantalla en el escritorio para la visualización del archivo.

Segmento de código 2.5. Comportamiento de botones en la interfaz de presentación

```
% Button pushed function: archivoButton
function archivoButtonPushed(app, event)
    appgeneral1
    delete(app)
end

% Button pushed function: archivosButton
function archivosButtonPushed(app, event)
    appgeneral2
    delete(app)
end
```

Ahora se procede a la implementación de la interfaz que muestra las gráficas para un archivo. Inicialmente se realiza la codificación del comportamiento en el cierre de la ventana

mediante la función *delete (app)* y de manera seguida se llama a la apertura de la interfaz de presentación para elegir nuevamente la opción de uno o dos archivos mediante la línea *appintro*, tal y como se presenta el segmento de código 2.6.

Segmento de código 2.6. Comportamiento del cierre de la interfaz para un archivo

```
% Close request function: UIFigure
function UIFigureCloseRequest(app, event)
    delete(app)
    appintro
end
```

Considerando la figura 2.5 donde se presentó el bosquejo de los botones, cajas de información, espacios de edición y axis de gráfica, se divide el código en función de todos estos elementos.

2.3.2.1 Botón cargar archivo

Inicialmente se declara las variables globales para su posterior uso dentro de los demás botones tal y como se presenta en el segmento de código 2.7

- A: Representa al archivo telemétrico.
- t_min: Valor del tiempo mínimo del que se tiene información en el presente archivo.
- t_max: Valor del tiempo máximo del que se tiene información en el presente archivo.
- graf1: Valor para activación de la primera gráfica.

Segmento de código 2.7. Variables globales del botón cargar archivo

```
global A t_min t_max graf1 % Se declara variables globales
```

En el segmento de código 2.8 se presenta el algoritmo para cargar el archivo telemétrico, de inicio se hace uso de la función *uigetfile* que permite la apertura del explorador de archivos para una fácil búsqueda del mismo, de tal manera que se devuelve el nombre del archivo y su dirección dentro del equipo. Inmediatamente se procede a importar únicamente los sub-archivos que interviene en lista gráficas propuesta, mediante la función *load* y se lo va concatenando en la estructura global A.

Después se procede a mostrar el nombre del archivo seleccionado actualmente en el área de texto denominada *app.TextArea.Value* que corresponde a la caja de información *Nombre de archivo cargado* de la figura 2.5.

Segmento de código 2.8. Cargar archivo

```
[file,path,indx] = uigetfile
filename = [path file]

A.RAD=load(filename,'RAD'); %Extraccion de sub-archivo RAD
A.RAD = A.RAD.RAD;
A.GPS=load(filename,'GPS');%Extraccion de sub-archivo GPS
A.GPS = A.GPS.GPS;
A.ATT=load(filename,'ATT');%Extraccion de sub-archivo ATT
A.ATT = A.ATT.ATT;
A.ORGN=load(filename,'ORGN');%Extraccion de sub-archivo ORGN
A.ORGN = A.ORGN.ORGN;
A.BARO=load(filename,'BARO');%Extraccion de sub-archivo BARO
A.BARO = A.BARO.BARO;

app.TextArea.Value = file;
```

Posteriormente, es necesario la extracción del tiempo general, el cual puede ser obtenido de cualquiera de los sub-archivos, es por ello que se ha generalizado la obtención del tiempo desde el sub-archivo “RAD”, debido a que es donde más se encuentran parámetros que influyen en la calidad del enlace.

En esta sección es necesario recordar, que cada columna de un sub-archivo está ligado a un parámetro en específico por lo que se ha identificado como la columna 2 para el tiempo y la columna 3 para el RSSI.

Se nota que, los valores del RSSI vienen en su propia escala por lo que es necesario su equivalencia en dBm, para ello se emplea la fórmula de conversión y se guarda los datos en un nuevo vector *RSSIdBm*, como se muestra en el segmento de código 2.9.

Segmento de código 2.9. Extracción de datos de tiempo y RSSI

```
tiempo_general = A.RAD(:,2); %Extraccion del tiempo general del sub-archivo RAD
RSSI=A.RAD(:,3); %extraccion de datos del RSSI
RSSIdBm = (RSSI/1.9)-127;
```

A continuación, en el segmento de código 2.10 se realiza la extracción el tiempo mínimo y máximo que se tiene en el vector de datos de *tiempo_general*. De manera seguida se

redondea estos valores, con el objetivo de presentar información de tiempo más significativa para el usuario, es por ello que también se realiza la conversión del tiempo a minutos ya que inicialmente dentro de la extracción de los archivos se tiene la información en microsegundos.

Una vez obtenidos los valores, se configura los límites en la caja de edición de intervalo de tiempo para que se dé una restricción al usuario y no inserte valores por fuera del mismo. De esta manera se evita errores en la ejecución.

Segmento de código 2.10. Ajuste de límite de tiempo en minutos

```
format short
%Redondeo del valor del tiempo minimo y maximo a 2 decimales
t_min = round(min(tiempo_general),2);
t_max = round(max(tiempo_general),2);
%Tiempo minimo y maximo en minutos
R_minu_t_min=round(t_min*1.6667e-8,2);
R_minu_t_max=round(t_max*1.6667e-8,2);

format short
%Se establece el minimo y maximo en la caja de edicion del intervalo de tiempo

app.TiempoInicialEditField_num.Limits = [R_minu_t_min ,R_minu_t_max];
app.TiempoFinalEditField_num.Limits = [R_minu_t_min ,R_minu_t_max];
% Se asigna el valor del intervalo de tiempo actual en la caja de edicion
app.TiempoInicialEditField_num.Value =R_minu_t_min;
app.TiempoFinalEditField num.Value =R_minu_t_max;
```

En el segmento de código 2.11 se presenta el tiempo máximo y mínimo en el área de texto *app.TextArea_infolimits* que corresponde a la caja *información de límites de tiempo* de la figura 2.6. Esto se realiza con el objetivo de presentar al usuario el tiempo en el que se puede escoger un intervalo para la gráfica.

Segmento de código 2.11. Presentación de tiempo

```
format short
% Se muestra el valor del intervalo de datos en la caja de informacion de limites de tiempo
app.TextArea_infolimits.Value = "En el intervalo de datos, el tiempo minimo es: " ...
+ num2str(R_minu_t_min) + " ; tiempo maximo es : " + num2str(R_minu_t_max);
```

Un parámetro necesario para una de las gráficas es el vector con los datos de distancia entre la estación terrena y el UAV en función del tiempo. Para su cálculo se sigue el proceso mencionado en la fase de diseño.

En el segmento de código 2.12, se procede a la ubicación de los datos de Latitud y Longitud del sub-archivo “GPS” y los datos de Altitud del sub-archivo “BARO” para su posterior extracción en función del tiempo general mediante las funciones *igualtiempos* y *selecposicion*.

Segmento de código 2.12. Extracción de datos de longitud, latitud y altitud

```
%extraccion de tiempos del log GPS en base al tiempo general
posGPS=igualtiempos(tiempo_general,A.GPS(:,2));
LATGPS = selecposicion(posGPS,A.GPS(:,8));
LNGGPS = selecposicion(posGPS,A.GPS(:,9));

%extraccion de tiempos de log BARO en base al tiempo general
posBARO=igualtiempos(tiempo_general,A.BARO(:,2));
ALTBARO = selecposicion(posBARO,A.BARO(:,3));
```

Una vez obtenidos los vectores con los datos ordenados, se procede a la obtención del arco formado entre los dos puntos, en unidad de grados. Para ello se hace uso de la función *distance* del Mapping Toolbox en Matlab que permite obtener varios datos de distancia entre dos puntos en una esfera, tal y como se observa en el segmento de código 2.13.

Segmento de código 2.13. Cálculo del arco entre dos puntos GPS

```
% Calculo del arco entre los puntos GPS
for i=1:length(posGPS)
[arclen(i),az(i)]=distance('gc',LATGPS(i),LNGGPS(i),A.ORG(1,4),A.ORG(1,5));
end
```

A continuación, en el segmento de código 2.14 se calcula la distancia en línea recta con la información del arco con la ayuda de la función *sind* que se refiere a la función trigonométrica seno en grados dentro de Matlab y con el uso del radio de la tierra en metros.

Segmento de código 2.14. Cálculo distancia en línea recta entre dos puntos GPS

```
%distancia en LINEA RECTA GPS en metros
r_tierra = 6378 * 1000 ; %radio de la tierra
d_lr = 2* (r_tierra*sind(arclen/2));
```

A continuación, se presenta la figura 2.10 donde se puede entender de mejor la manera la forma en la que se realizó el cálculo, notando el resultado final *d_lr* que corresponde al vector de distancias en línea recta entre dos posiciones GPS en metros.

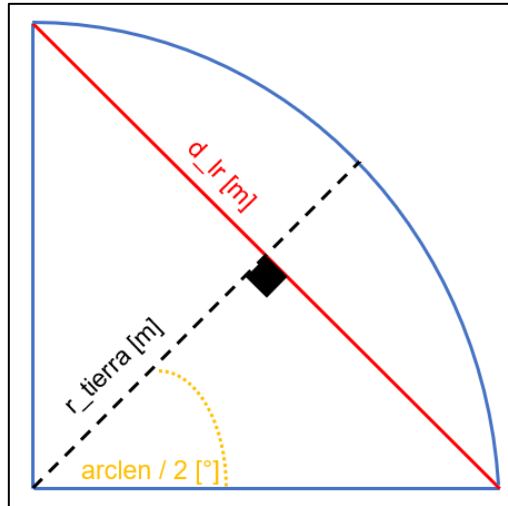


Figura 2.10. Representación de la distancia en línea recta entre dos posiciones GPS

Siguiendo con el cálculo de la distancia entre el UAV y la estación terrena, en el segmento de código 2.15 se hace uso del vector de distancias en línea recta con el vector de altitudes de tal manera que aplicando el teorema de Pitágoras se puede obtener la distancia total. Para ello se emplea la función *hypot* que permite hacer el cálculo de manera sencilla. Finalmente, se guarda las distancias en el vector *distanciatotal*.

Segmento de código 2.15. Cálculo distancia total UAV-Base

```
for j=1: length(d_lr)
distanciatotal(j) = hypot(d_lr(j),ALTBARO(j)); % distancia entre el UAV y estacion terrena
end
```

Tener en cuenta que la obtención de estos vectores de datos telemétricos en esta sección del código se realiza con el objetivo de inmediatamente se cargue el archivo mostrar la primera gráfica en la lista es decir la gráfica “Distancia entre UAV-BASE vs RSSI”. Por ello en el segmento de código 2.16 se configura de inicio el encendido del primer botón de la lista de gráficas *app.DistanciaentreUAVBASEvsRSSIButton* y se procede a mostrar la gráfica mediante la función *graficar*. Luego se asegura que se mantenga el primer botón de la lista de gráficas encendida mediante la línea *graf1=1*.

Segmento de código 2.16. Gráfica “Distancia entre UAV-BASE”

```
app.DistanciaentreUAVBASEvsRSSIButton.Value =1; %configuracion del grupo de botones de la lista
en la primera grafica
graficar(app.UIAxes,app.MximoTextArea,app.MnimoTextArea,app.PromedioTextArea,...
tiempo_general,distanciatotal,RSSIdBm,"Distancia","[m]","RSSI","[dBm]","Distancia entre UAV-BASE
vs RSSI");
graf1=1 %configuracion del grupo de botones de la lista en la primera grafica;
```

2.3.2.2 Grupo de botones lista de gráficas

Inicialmente en el segmento de código 2.17, se declara las variables globales de igual forma como se dio en el segmento de código 2.7. Como se puede observar se usa la variable *A* que es donde se guardan todos los parámetros del archivo log, como ya se había mencionado anteriormente. De igual forma se declara las variables graf1-graf6 que corresponden a la activación de las gráficas de tal manera que cuando alguna se encuentre en valor igual 1, se activara el botón que corresponde a cada gráfica. De manera seguida se asigna el valor de activación de cada gráfica a cada variable graf1-6 según corresponda, con el objetivo de utilizar dichas variables de manera más sencilla en el código.

Segmento de código 2.17. Lista de gráficas

```
global A graf1 graf2 graf3 graf4 graf5 graf6
graf1=app.DistanciaentreUAVBASEvsRSSIButton.Value;
graf2=app.PrdidasenelespaciolibrevsRSSIButton.Value;
graf3=app.ROLLvsRSSIButton.Value;
graf4=app.PITCHvsRSSIButton.Value;
graf5=app.RemNoisevsRemRSSIButton.Value;
graf6=app.NoisevsRSSIButton.Value;
```

Se crea un nuevo vector tiempo general debido a que el usuario puede o no haber editado el espacio del intervalo de tiempo, es por ello que esta sección de código es importante, debido a que las gráficas se mostrarán en función de dicho tiempo. Para ello, es necesario que se obtenga el valor actual en minutos de la caja de edición del intervalo de tiempo y se lo convierta a microsegundos. Notar que se obtienen dos valores, el tiempo inicial y final configurados por el usuario. Luego, se realiza la búsqueda de dichos valores en el vector de tiempo general y se procede a recortar, desde el dato de tiempo 1 hasta el dato inicial dado por el usuario y desde el tiempo final dado por el usuario hasta el tiempo máximo dentro del vector tiempo general, tal y como se muestra en el segmento de código 2.18.

Segmento de código 2.18. Ajuste del vector de tiempo acorde un nuevo intervalo

```
%--NUEVO VECTOR TIEMPO GENERAL
%Obtencion de los tiempos inicial y final de las cajas edicion de intervalos y conversion a
microsegundos
t1= app.TiempoInicialEditField_num.Value/1.6667e-8;
t2= app.TiempoFinalEditField_num.Value/1.6667e-8;
%Busqueda de los tiempos inicial y final solicitados, dentro del vector tiempo_general
[ minimoabc(1), pos_limite(1) ] = min(abs(tiempo_general-t1));
[ minimoabc(2), pos_limite(2) ] = min(abs(tiempo_general-t2));
%Generacion del nuevo vector tiempo general
tiempo_general([pos_limite(2)+1:length(tiempo_general)]) = []; % Recorte de todos los tiempos
desde el maximo hasta el final
tiempo_general([1:pos_limite(1)]) = []; %Recorte de todos los tiempos desde el inicio hasta el
tiempo minimo
```

A continuación, en el segmento de código 2.19, se presenta la obtención del vector de pérdidas para la gráfica “pérdidas vs RSSI”. Es importante aclarar que la caja de edición de frecuencia es configurada con el valor inicial de 915 MHz, pero puede ser editada por el usuario a cualquier valor, es por ello que se captura la frecuencia de dicha caja de edición y se convierte en GHz para posteriormente ser usada en la fórmula para el cálculo de las pérdidas en el espacio libre para un UAV.

Se puede notar que dicho resultado también está en función del vector *distanciatotal* el cual corresponde a los datos de la distancia entre el UAV y la estación base, que se detalló en la sección 2.3.2.1.

Segmento de código 2.19. Cálculo de las pérdidas en el espacio libre

```
%-- Pérdidas vs RSSI
f = app.FrecuenciaEditField.Value / 1000; % Frecuencia en GHz
Lf = 92.45 + 20*log10(f)+20*log10(distanciatotal/1000); % Calculo de pérdidas en dB
```

A continuación, en el segmento de código 2.20, se presenta la obtención de los vectores de datos ROLL y PITCH, los cuales corresponden al sub-archivo ATT y de igual forma se hace uso de las funciones *igualtiempos* y *selecposicion* para dejarlos sincronizados en función del tiempo general actual.

Segmento de código 2.20. Extracción de los datos ROLL y PITCH

```
%--PITCH | ROLL vs RSSI
posATT=igualtiempos(tiempo_general,A.ATT(:,2)); %Igualacion de tiempos al sub-archivo ATT
ROLLATT = selecposicion(posATT,A.ATT(:,4)); %Extraccion de datos ROLL
PITCHATT = selecposicion(posATT,A.ATT(:,6)); %Extraccion de datos PITCH
```

A continuación, en el segmento de código 2.21, se presenta la obtención de los vectores de datos RemNoise, RemRSSI, Noise, los cuales corresponden al sub-archivo RAD y de igual forma se hace uso de las funciones *igualtiempos* y *selecposicion* para dejarlos sincronizados en función del tiempo general actual y de igual manera se realiza la conversión a dBm de cada uno de los vectores.

Segmento de código 2.21. Extracción de los datos RemNoise, RemRSSI y Noise

```
%-- RemNoise vs RemRSSI
posRAD=igualtiempos(tiempo_general,A.RAD(:,2)) %Igualacion de tiempos al sub-archivo RAD
RemNoise = selecposicion(posRAD,A.RAD(:,7)); %Extraccion de datos RemNoise
RemRSSI = selecposicion(posRAD,A.RAD(:,4)); %Extraccion de datos RemRSSI
RemRSSI = (RemRSSI/1.9)-127; %Conversion a dBm de los datos RemNoise
RemNoise = (RemNoise/1.9)-127;%Conversion a dBm de los datos RemNoise
%--NoisevsRSSI
Noise= selecposicion(posRAD,A.RAD(:,6)); %Extraccion de datos Noise
Noise = (Noise/1.9)-127; %Conversion a dBm de los datos Noise
```

Finalmente, en el segmento de código 2.22, se hace uso de la función *if*, que permite la selección de cada gráfica en función de cuál de los botones se encuentre activado en la lista de gráficas al momento. Como se puede observar en los espacios se aplica la función *graficar* para las gráficas que requieren la comparación de dos tipos de datos en unidades diferentes y la función *graficar2* para las gráficas que contienen la misma unidad de medida.

Segmento de código 2.22. Selección de gráficas

```

if graf1==1
graficar(app.UIAxes,app.MximoTextArea,app.MnimoTextArea,app.PromedioTextArea,...
tiempo_general,distanciatotal,RSSIdBm,"Distancia","[m]","RSSI","[dBm]","Distancia entre UAV-BASE
vs RSSI");

elseif graf2==1
graficar(app.UIAxes,app.MximoTextArea,app.MnimoTextArea,app.PromedioTextArea,...
tiempo_general,Lf,RSSIdBm,"Pérdidas","[dB]","RSSI","[dBm]","Pérdidas en el espacio libre vs
RSSI");

elseif graf3==1
graficar(app.UIAxes,app.MximoTextArea,app.MnimoTextArea,app.PromedioTextArea,...
tiempo_general,ROLLATT,RSSIdBm,"ROLL","[°]","RSSI","[dBm]","ROLL vs RSSI");

elseif graf4==1
graficar(app.UIAxes,app.MximoTextArea,app.MnimoTextArea,app.PromedioTextArea,...
tiempo_general,PITCHATT,RSSIdBm,"PITCH","[°]","RSSI","[dBm]","PITCH vs RSSI");

elseif graf5==1
graficar2(app.UIAxes,app.MximoTextArea,app.MnimoTextArea,app.PromedioTextArea,...
tiempo_general,RemNoise,RemRSSI,"RemNoise","RemRSSI","[dBm]","RemNoise vs RemRSSI");

elseif graf6==1
graficar2(app.UIAxes,app.MximoTextArea,app.MnimoTextArea,app.PromedioTextArea,...
tiempo_general,Noise,RSSI,"Noise","RSSI","[dBm]","Noise vs RSSI");

end

```

Ahora bien, viendo la necesidad de que se muestre la gráfica al instante que se realice el cambio por parte del usuario de los intervalos de tiempo o de la frecuencia, se incluye la misma programación dentro de las funciones de activación: *TiempoInicialEditField_numValueChanged* , *TiempoFinalEditField_numValueChanged* y *FrecuenciaEditFieldValuechanged*, que corresponden a las cajas de edición: *Tiempo Inicial*, *Tiempo Final* y *Frecuencia*.

Para concluir esta sección se aclara que para la creación de la interfaz para dos archivos es exactamente la misma implementación presentada para un archivo, con la diferencia

que ahora se realiza los mismos procesos, pero en dos archivos log diferentes al mismo tiempo.

Segmento de código 2.23. Carga de archivo log 1

```
function cargarButtonPushed(app, event)
global A t_min t_max graf1

[file,path,indx] = uigetfile;
filename = [path file]
A = importdata(filename);
app.TextArea.Value = file;
```

Segmento de código 2.24. Carga de archivo log 2

```
function cargar_2ButtonPushed(app, event)
global B t_min2 t_max2 graf1

[file,path,indx] = uigetfile;
filename = [path file]
B = importdata(filename);
app.TextArea_2.Value = file;
```

Es decir que ahora se tienen 2 secciones para cargar archivos de tal manera que se guarda al archivo log 1 en la variable global A como se puede observar en la sección de código 2.23 y al archivo log 2 en la variable global B como se puede observar en la sección de código 2.24. De tal manera que se sigue el mismo proceso de obtención de datos de tiempo, parámetros, cálculos y proyección de gráficas.

3. RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1 RESULTADOS

3.1.1 PRUEBAS DE FUNCIONAMIENTO

Inicialmente para realizar las pruebas en la presente aplicación es necesario el uso de un archivo log en formato de contenedor de datos binarios de MATLAB (.mat), tal y como se mencionó en los requerimientos . Por lo que a continuación se presenta el procedimiento para obtener dicho archivo con la ayuda del software de simulación Mission Planner.



Figura 3.1. Ventana de Mission Planner | Data Flash Logs

Como se puede observar en la figura 3.1, se presenta la interfaz de Mission Planner ubicado en la pestaña Data Flash Logs. La cual ofrece gran variedad de opciones para el tratamiento de archivos tipo logs. A continuación, se debe escoger la opción *Create Matlab File*, la cual va a permitir convertir el archivo log en formato (.mat).

Una vez que se obtenga el archivo (.mat) se puede empezar a utilizar la aplicación. Como se detalló en la fase de diseño, inicialmente la aplicación contará con una ventana de presentación, la cual se muestra en la figura 3.2.



Figura 3.2. Ventana de presentación de la aplicación

El funcionamiento es sencillo, únicamente se presenta el título del proyecto y la opción para elección de uno o dos archivos, los cuales saltan a una siguiente ventana, que se va a detallar a continuación.

3.1.1.1 Aplicación para procesamiento de un archivo

Una vez seleccionado la opción para un archivo, se presenta la interfaz de la figura 3.3 en la cual se despliega una ventana a pantalla completa, que incluye las secciones de carga de archivo, selección de gráfica, edición de intervalos de tiempo, edición de frecuencia, información de tiempo, nombre de archivo, valor máximo, mínimo y promedio, además de su correspondiente espacio para la proyección de la gráfica.

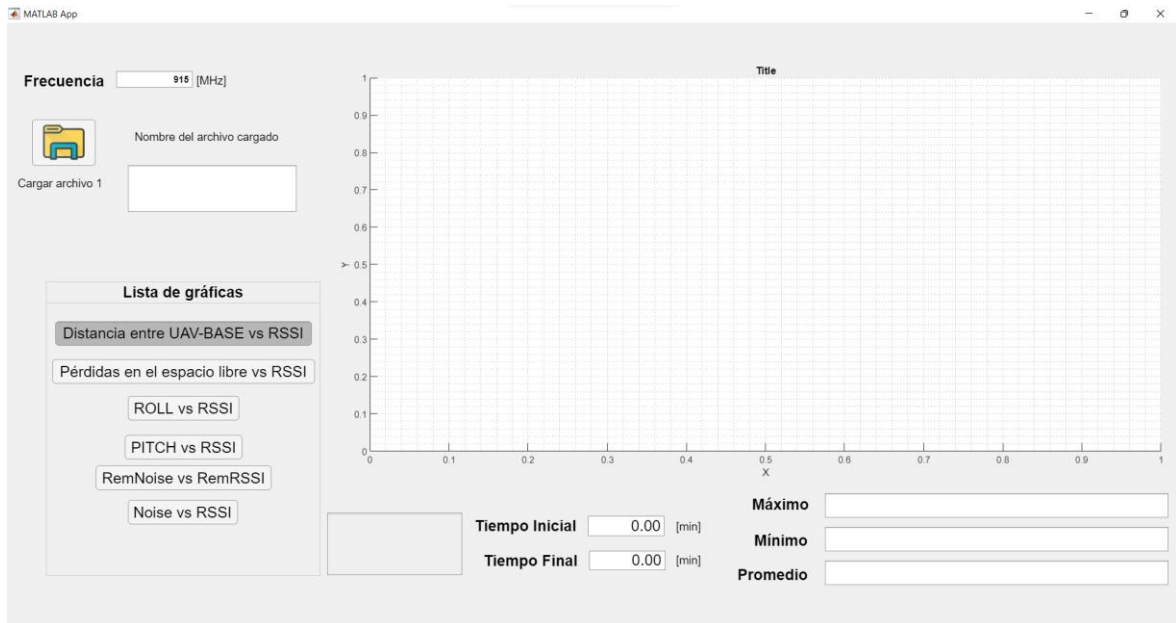


Figura 3.3. Ventana de la interfaz para procesamiento de un archivo.

A continuación, se detalla las pruebas de funcionamiento de cada una de las funciones.

3.1.1.1.1 Cargar archivo

Como se puede observar en la figura 3.4, cuando se selecciona el botón de carga de archivo 1, se despliega el explorador de archivos para poder elegir el archivo log y de esta manera cargarlo a la aplicación.

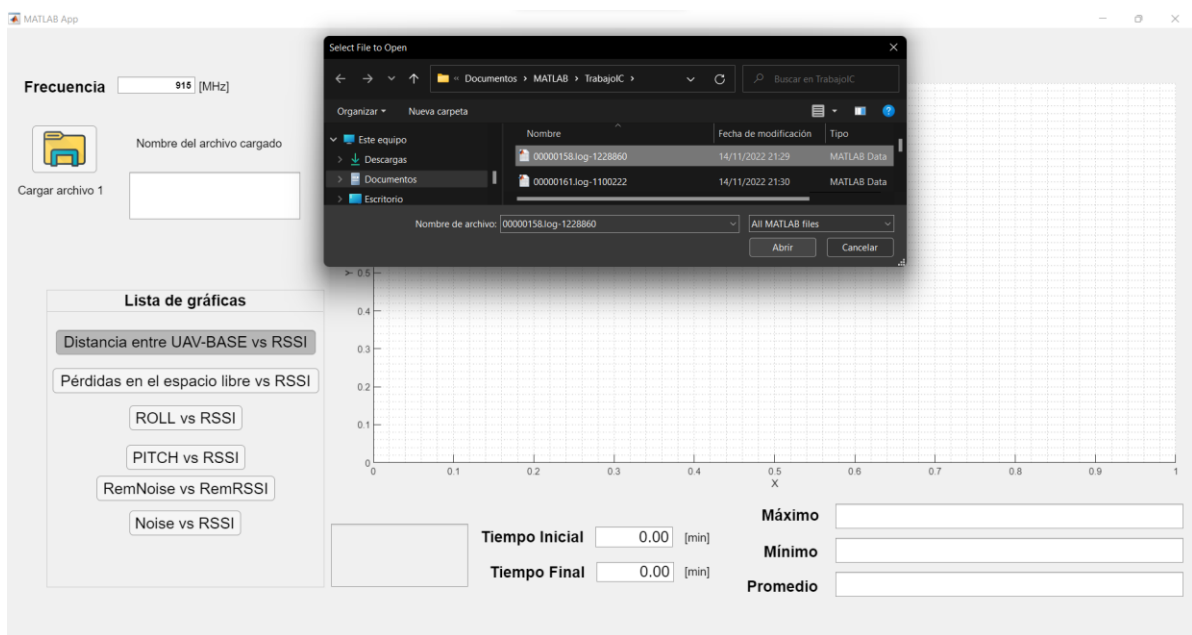


Figura 3.4. Funcionalidad de carga de archivo

De manera seguida, se muestra el nombre del archivo cargado actualmente y se proyecta la primera gráfica en la lista, tal y como se observa en la figura 3.5. Es importante notar la información que se despliega en las áreas de texto, por ejemplo, se observa los límites de tiempo, en este caso que va de 3.06 a 37.18 minutos y entre los cuales se puede seleccionar un intervalo por el usuario. Además, se incluyen los valores máximos, mínimos y promedios de los datos dentro de la gráfica, que este caso son la distancia en metros y el RSSI en dBm.

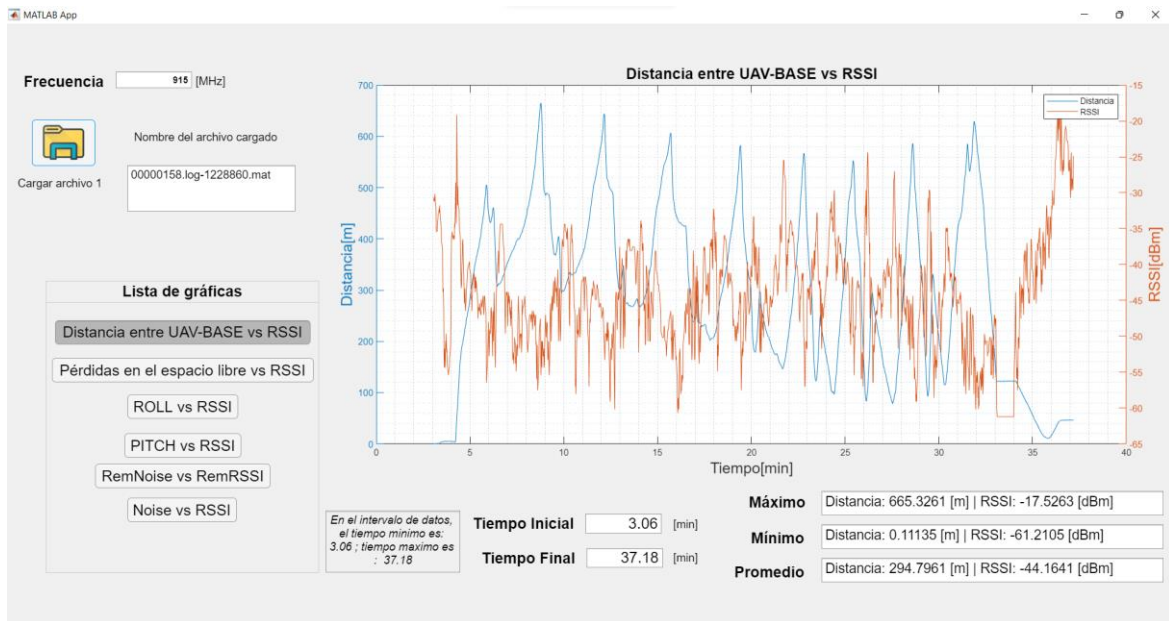


Figura 3.5. Proyección de la gráfica “Distancia entre UAV-BASE vs RSSI”

3.1.1.1.2 Selección de gráfica

Dentro de la lista, únicamente se elige el botón con el título de la gráfica que se desea visualizar y de manera seguida se proyecta los diferentes parámetros, de tal manera que se muestren correctamente: el título de la gráfica, leyendas, la etiqueta en el eje “x” y las etiquetas en el eje “y”, que como se explicó en la fase de diseño, en los 4 primeros casos de la lista de gráficas se tiene la comparación de dos parámetros en unidades de medida diferentes, por lo que se muestran dos ejes “y” a diferente escala, que en este caso son las pérdidas en el espacio libre en dB al lado izquierdo y el RSSI en dBm al lado derecho, como se puede observar en la figura 3.6. De igual forma se tiene la información de valores máximos, mínimos y promedios.

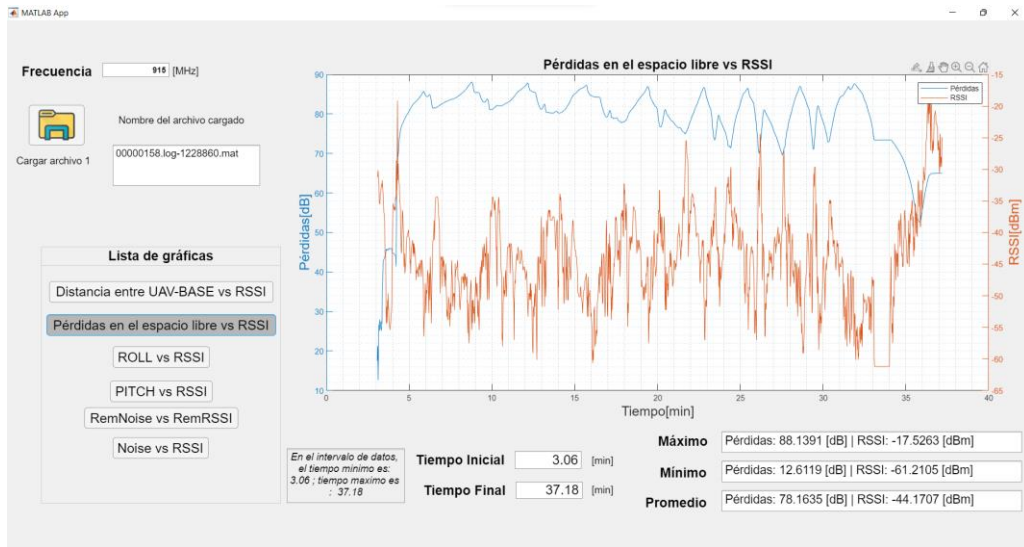


Figura 3.6. Proyección de la gráfica “Pérdidas en el espacio libre vs RSSI”

La figura 3.7 se puede contrastar con la figura 3.6 debido a que se selecciona la gráfica que incluye dos parámetros en las mismas unidades, como se puede observar se tiene un único eje “y” a una misma escala, en el que se muestran los dos datos telemétricos, en este caso el Noise y RSSI ambos en dBm.

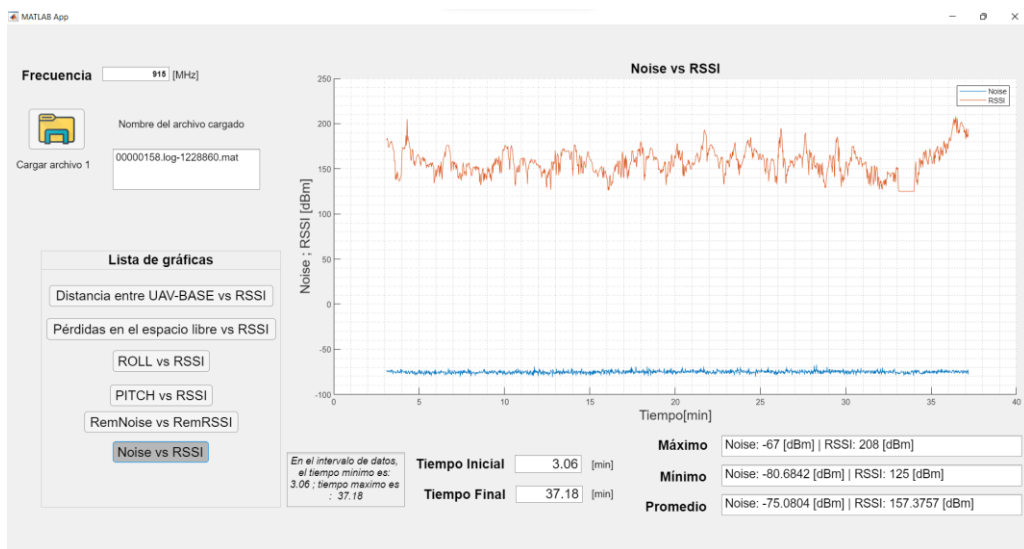


Figura 3.7. Proyección de la gráfica “Noise vs RSSI”

3.1.1.1.3 Cambio de intervalo de tiempo

Como se muestra en la figura 3.8, se realiza un cambio en el intervalo de tiempo por parte del usuario que este caso va de 10.32 a 20.15 minutos, notar que se puede incluir hasta dos decimales en la selección de tiempo. De manera seguida se despliega la gráfica con los nuevos límites y además notar que se muestran los valores de máximos, mínimos y promedios, pero en esta ocasión dentro del nuevo intervalo.

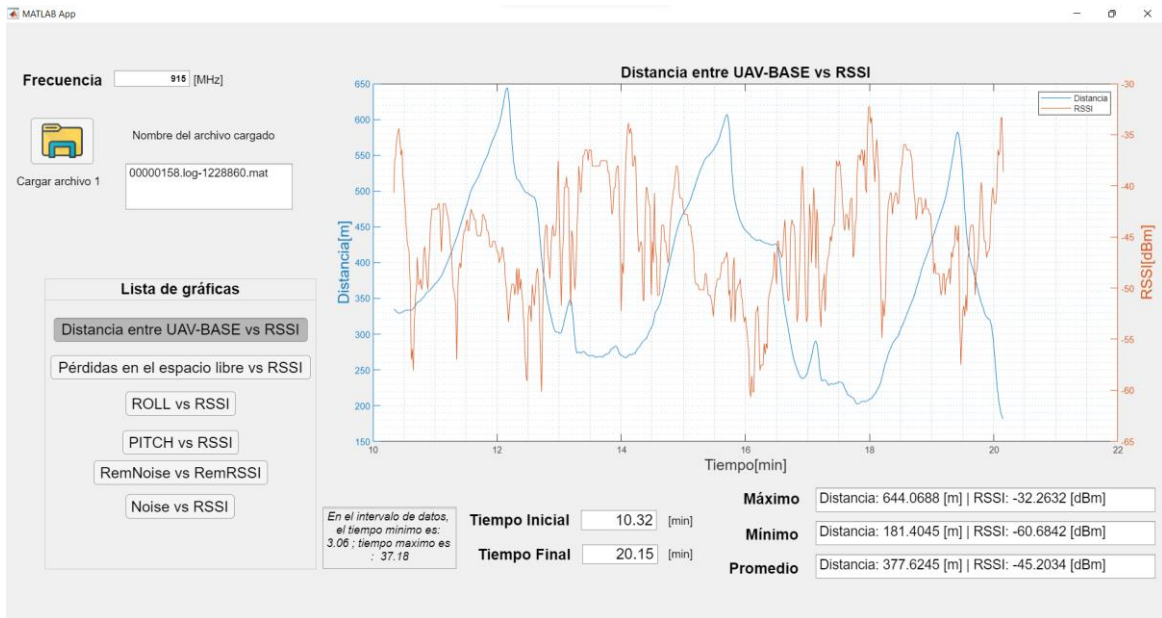


Figura 3.8. Cambio del intervalo de tiempo

3.1.1.2 Aplicación para procesamiento de dos archivos

Una vez seleccionada la opción para dos archivos, se presenta la interfaz de la figura 3.9 en la cual se despliega una ventana a pantalla completa, que incluye las secciones de carga de dos archivos, selección de gráfica, edición de dos intervalos de tiempo, edición de frecuencia, información de dos nombres de archivo, valores máximos, mínimos y promedios, teniendo en cuenta su correspondiente espacio para la proyección de dos gráficas de manera simultánea.

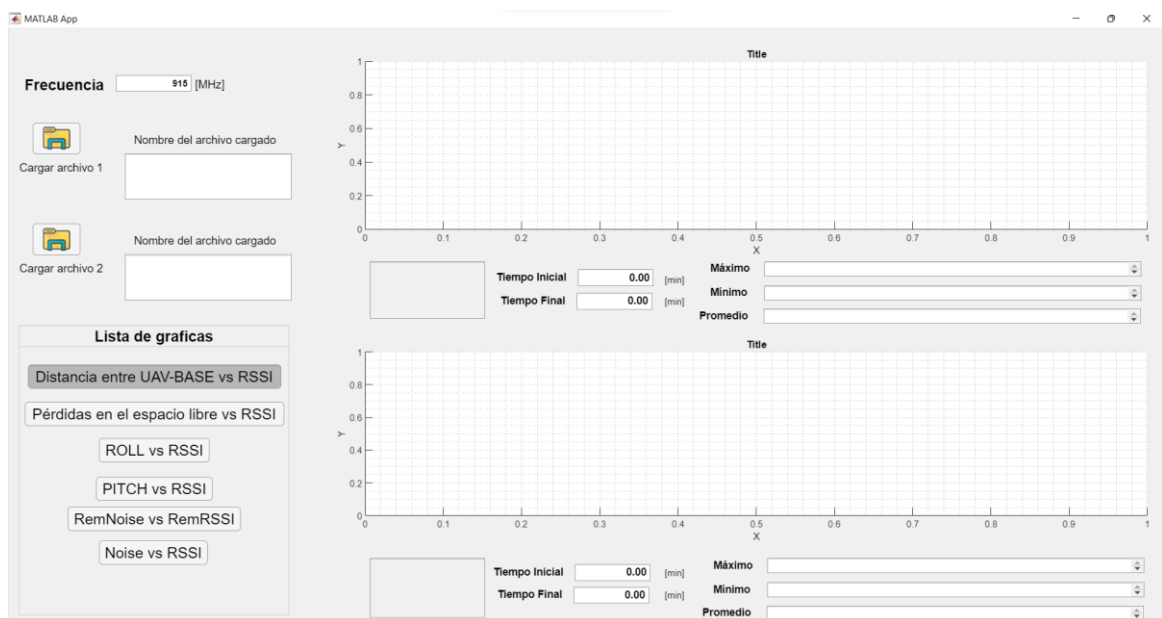


Figura 3.9. Ventana de la interfaz para procesamiento de dos archivos.

A continuación, se detalla las pruebas de funcionamiento de cada una de las funciones.

3.1.1.2.1 Cargar archivo

Como se puede observar en la figura 3.10, se puede realizar la carga de dos archivos en simultáneo, teniendo en cuenta que apenas se cargue el primer archivo se desplegará la primera gráfica en la parte superior. De igual manera ocurre al cargar el segundo archivo, como se puede observar en la figura 3.11. En este ejemplo se presentan dos archivos log diferentes.

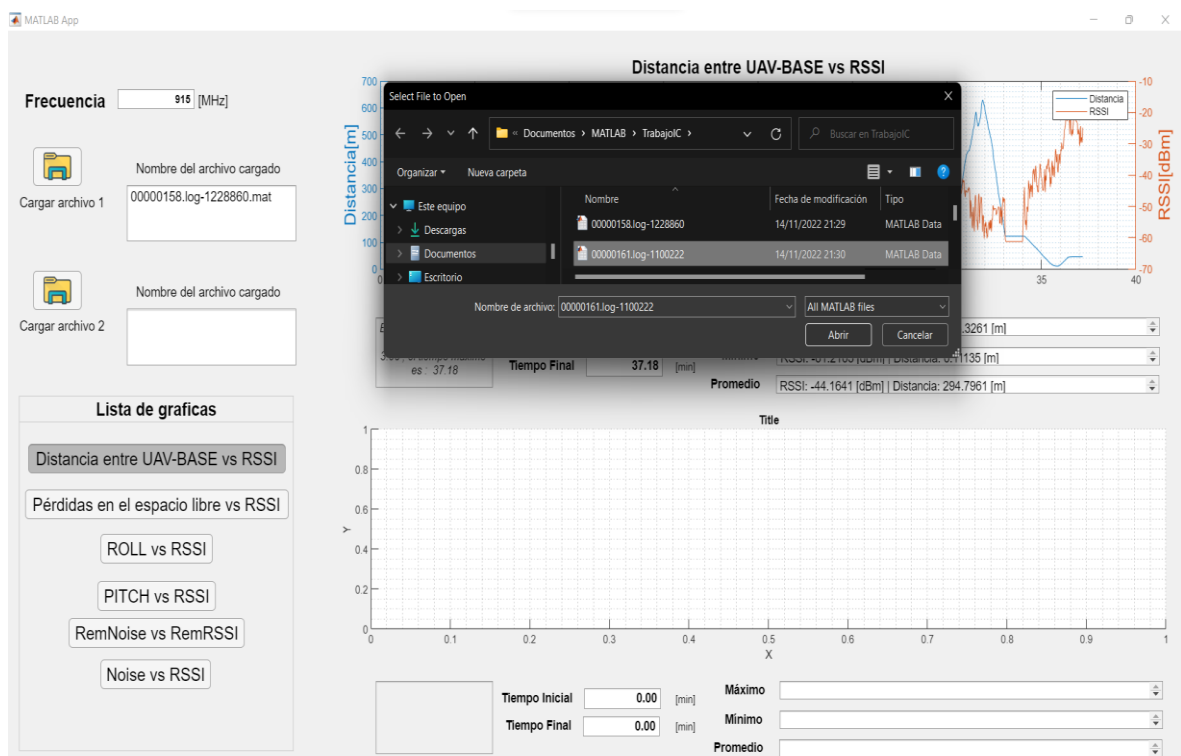


Figura 3.10. Funcionalidad de carga de dos archivos

Como se observa en la figura 3.11, se muestran los nombres de los archivos cargados actualmente y se proyecta la primera gráfica en la lista para los dos casos.

De igual manera es importante notar la información que se despliega en las áreas de texto, por ejemplo, se observa los límites de tiempo, para el archivo 1 que va de 3.06 a 37.18 minutos y para el archivo 2 que va de 3.77 a 34.1 minutos entre los cuales se puede seleccionar un intervalo por parte del usuario. Además, se incluye los valores máximos, mínimos y promedios de los datos dentro de las gráficas, que este caso son la distancia en metros y el RSSI en dBm.

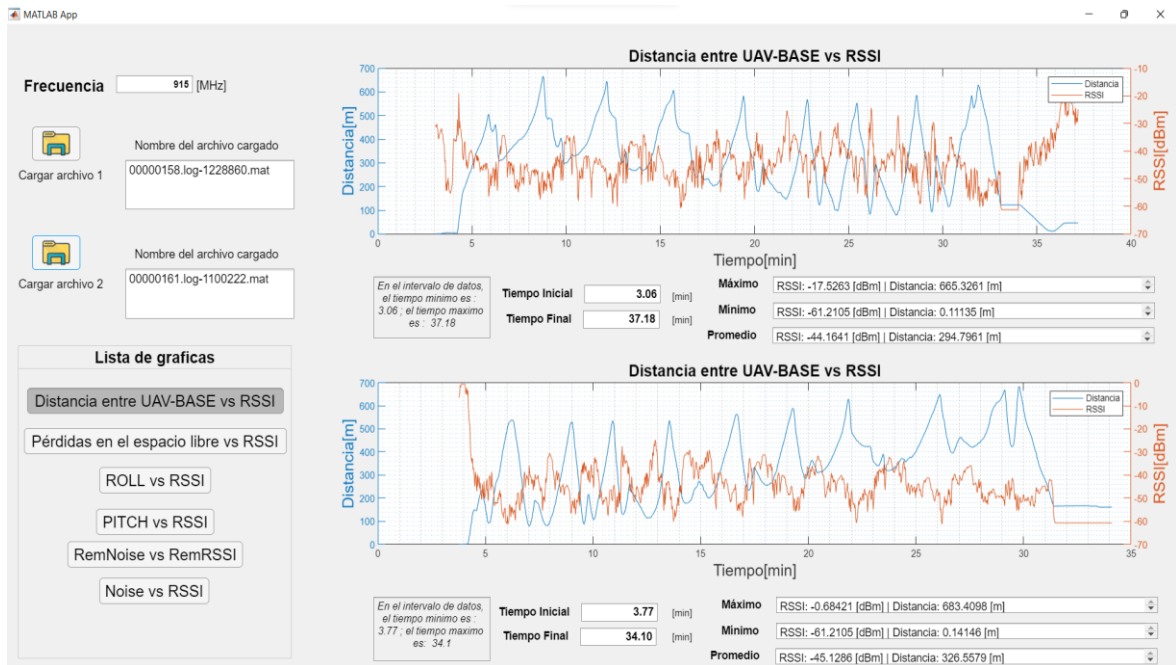


Figura 3.11. Proyección de la gráfica “Distancia entre UAV-BASE vs RSSI” para dos archivos

Las siguientes funcionalidades corresponden a la selección de gráfica y cambio de intervalo, que se explicó en la sección anterior, por lo que en este punto se aclara que se trata del mismo funcionamiento, pero en esta ocasión se lo realiza para dos archivos por separado, proyectando los resultados en una misma ventana.

3.1.2 ANÁLISIS DE ARCHIVOS Y GRÁFICAS

Habiendo realizado las pruebas de funcionamiento, se procede a la evaluación de las gráficas resultantes a fin de determinar correlaciones de los parámetros seleccionados en la fase de diseño, con el objetivo de visualizar patrones de comportamiento entre ellos.

A continuación, se incluye una sección que describe los vuelos que corresponden a cada archivo log, con el objetivo de tener en cuenta los diferentes comportamientos de vuelo.

3.1.2.1 Descripción de archivo log

El plan de vuelo se realizó en el humedal Pugllhuma de la reserva ecológica Antisana, ubicado en las provincias de Pichincha y Napo en Ecuador.

- Archivo 0000158 : se trata de un vuelo tipo mapeo, es decir se sigue una trayectoria de puntos previamente configurados, con una duración de 20 minutos

aproximadamente, el vuelo se lo realiza sin cámara, pero con un peso equivalente de 250g en reemplazo de la cámara para equilibrar el centro de gravedad.

3.1.2.2 Análisis de gráficas

Para el análisis de las gráficas se toma de muestra el archivo 0000158. Se inicia con la primera gráfica que corresponde a la distancia entre UAV y la estación base vs el RSSI, tal y como se muestra en la figura 3.12.

Se nota que existe una relación inversa entre la distancia y el RSSI, de tal manera que se puede observar en la gráfica que a medida que se aumenta la distancia se va reduciendo el RSSI, es decir se va a reducir el nivel de potencia de la señal. Inclusive se puede notar que en algunos instantes, los picos máximos de distancia están sincronizados con los picos mínimos de RSSI o viceversa.

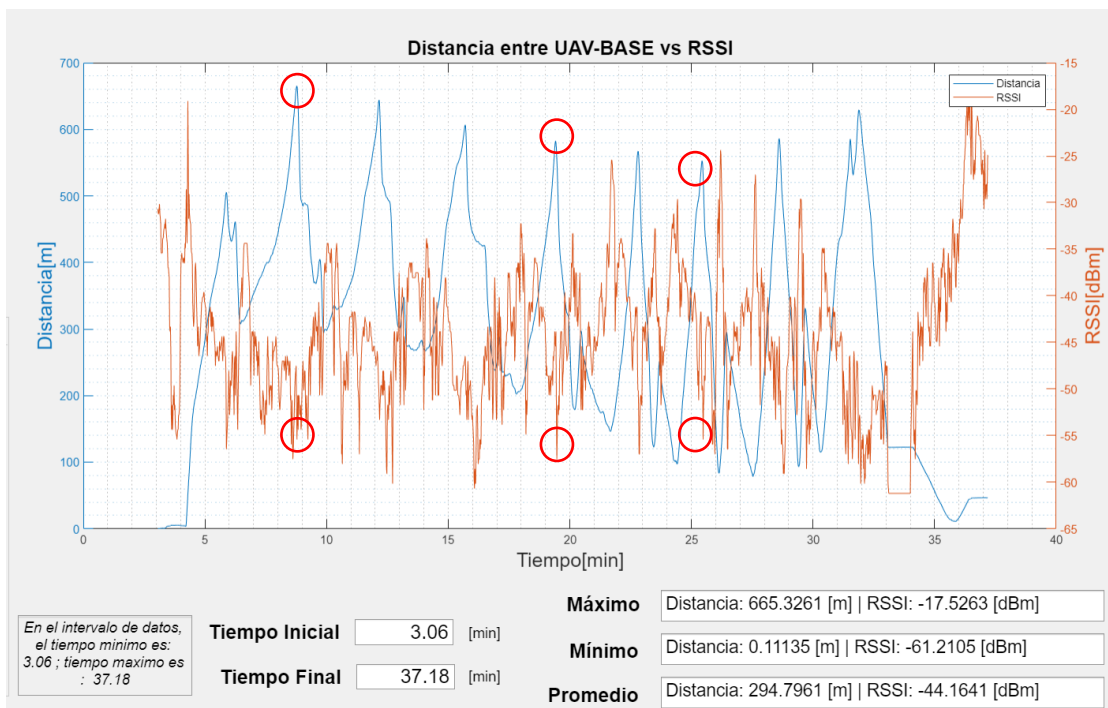


Figura 3.12. Distancia entre UAV-BASE vs RSSI | Archivo: 0000158

En realidad, son resultados que ya se esperaban debido a que a medida que incrementa la distancia, se incrementa las pérdidas en la comunicación por lo que a continuación se presenta la gráfica de pérdidas en el espacio libre vs el RSSI en la figura 3.13 donde de igual manera se tiene una relación inversa. Aunque no es tan notable como en la figura 3.12 existen algunos puntos donde si se puede notar dicha relación.

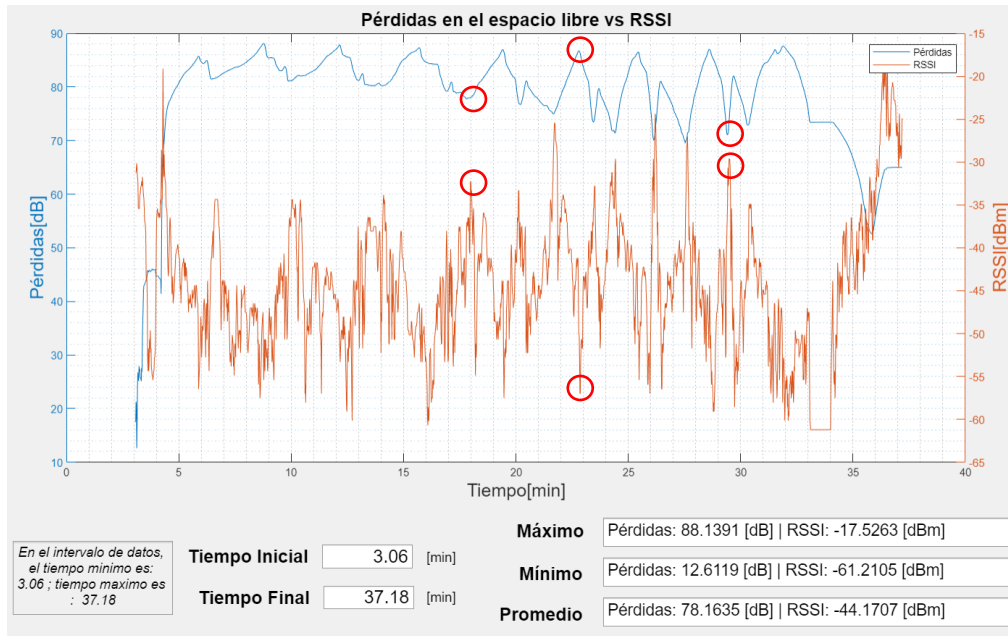


Figura 3.13. Pérdidas en el espacio libre vs RSSI | Archivo: 0000158

Otro de los parámetros importantes se basa en el RSSI en función de como gira el UAV, teniendo en cuenta que debido al movimiento se pierde la alineación de las antenas, es por ello que a continuación se presenta el análisis de las gráficas ROLL vs RSSI y PITCH vs RSSI. Como se observa en la figura 3.14, no es posible distinguir ningún patrón de comportamiento sólido, sino más bien algunos puntos donde se indica que al existir un mayor giro de roll en positivo se va perdiendo un poco la señal, aun así, la evidencia no es suficiente como para inferir una relación de comportamiento entre el uno y el otro.

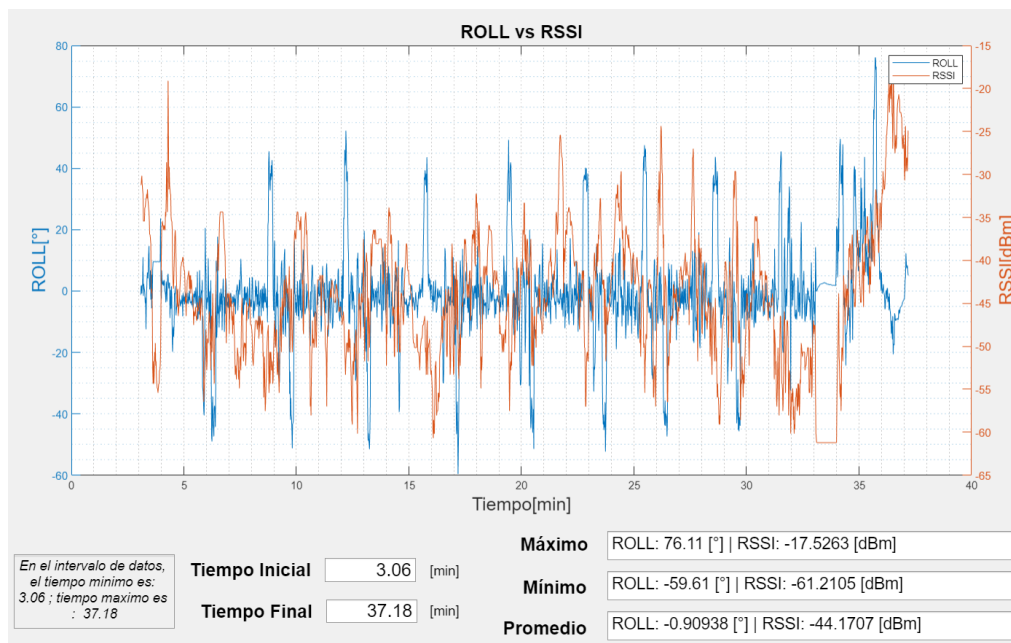


Figura 3.14. ROLL vs RSSI | Archivo: 0000158

De igual manera en la figura 3.15 que representa la gráfica de PITCH vs RSSI no es posible encontrar alguna correlación importante entre las líneas. Por lo que finalmente se infiere que las pérdidas ocasionadas por la desalineación de la polarización de las antenas, no es tan representativa dentro de un enlace de telemetría en el presente plan de vuelo, como lo serían otros parámetros como por ejemplo la distancia.

Aun así, para tener mejores resultados, se podría realizar un plan de vuelo en el que se mantenga la distancia, lo cual se podría lograr en UAV tipo cuadricóptero en el que únicamente se varíe el movimiento en los ejes, pitch y roll, de tal manera que los resultados del RSSI no se vean influenciados por el factor distancia.

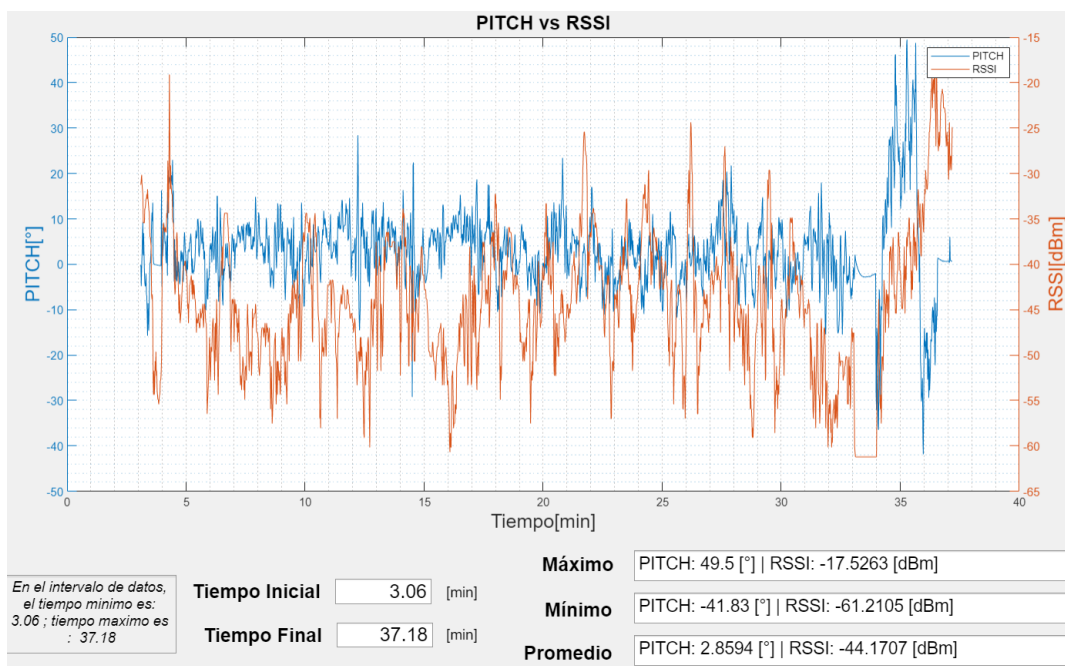


Figura 3.15. PITCH vs RSSI | Archivo: 0000158

A continuación, se presenta las gráficas del RemNoise vs RemRSSI y Noise vs RSSI en la figura 3.16 y 3.17 . Como se mencionó en el capítulo anterior los parámetros “Rem” se refieren a los datos tomados en el otro extremo de la comunicación, en este caso el archivo log es muestreado en la controladora de vuelo incluida en el UAV, es decir los datos “Rem” son los datos registrados en el lado de la estación base.

El parámetro RemNoise, que se refiere al piso de ruido registrado en el lado de la estación base, el cual casi siempre se encuentra por debajo del RSSI con un margen de al menos 10 dBm para la figura 3.16, lo que implica que existe comunicación en casi todo momento de una manera eficiente.

En este punto cabe mencionar que existe un pequeño tramo al final en donde la línea RemRSSI cae por debajo de la línea del RemNoise, lo que indica que se perdió la comunicación en dicho intervalo de tiempo.

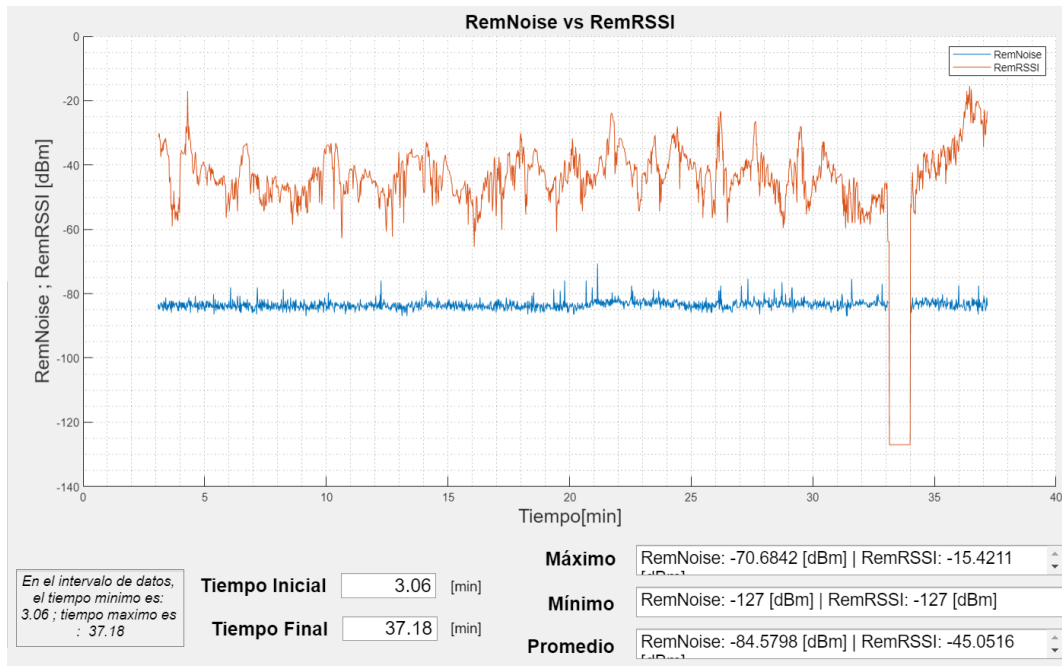


Figura 3.16. RemNoise vs RemRSSI | Archivo: 0000158

De manera similar en la figura 3.17, se observa que el RSSI siempre se encuentra por encima del piso de ruido con un margen mucho más grande de al menos 150 dBm lo que indica claramente que se tiene comunicación en todo momento.

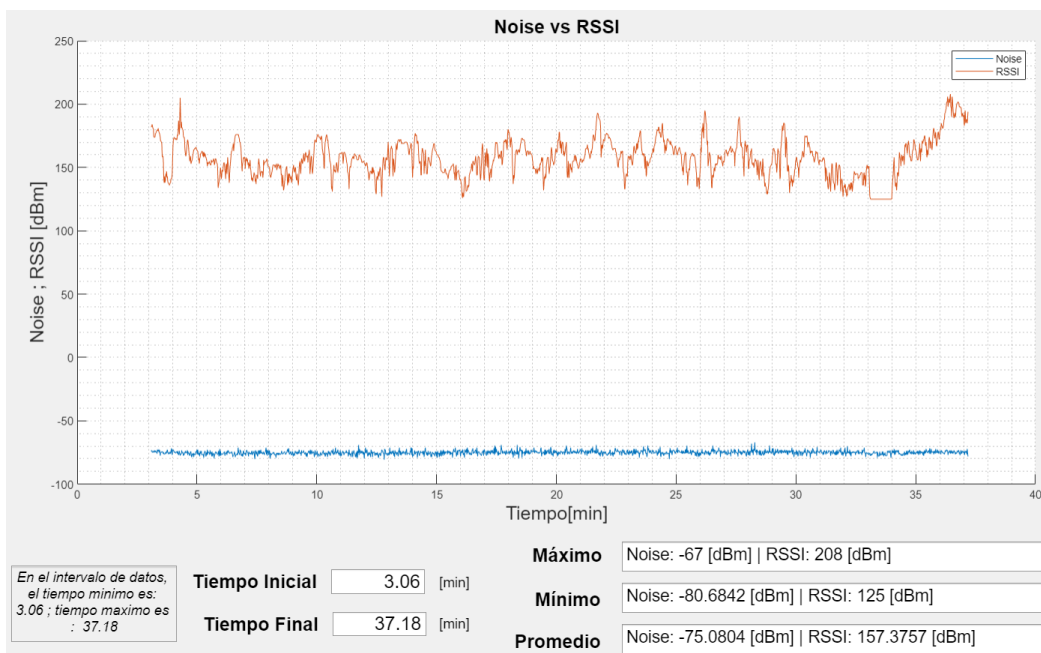


Figura 3.17. Noise vs RSSI | Archivo: 0000158

3.1.2.3 Contraste con visualizador de Mission Planner

Como se puede observar en la figura 3.18, en la parte superior se muestra el visualizador del Mission Planner y en la parte inferior se muestra la aplicación implementada, proyectando en cada uno el mismo archivo y mismos parámetros con el objetivo de poder visualizar que efectivamente se tiene la misma forma de las líneas y comprobar de esta manera que se están graficando de manera correcta. Ahora bien, es importante mencionar que los valores no coinciden, debido a que se realizó la conversión de los datos RSSI en unidades de dBm en la aplicación. Y también, se nota que la línea de tiempo no es la misma debido a que, en el visualizador de Mission Planner se tiene el eje “x” ajustado al tiempo en horas minutos y segundos en el día en el que se tomó los datos, mientras que en la aplicación se muestra únicamente los valores en minutos desde el primer registro hasta el último independientemente de la hora del día en el que se ejecutó el plan de vuelo.

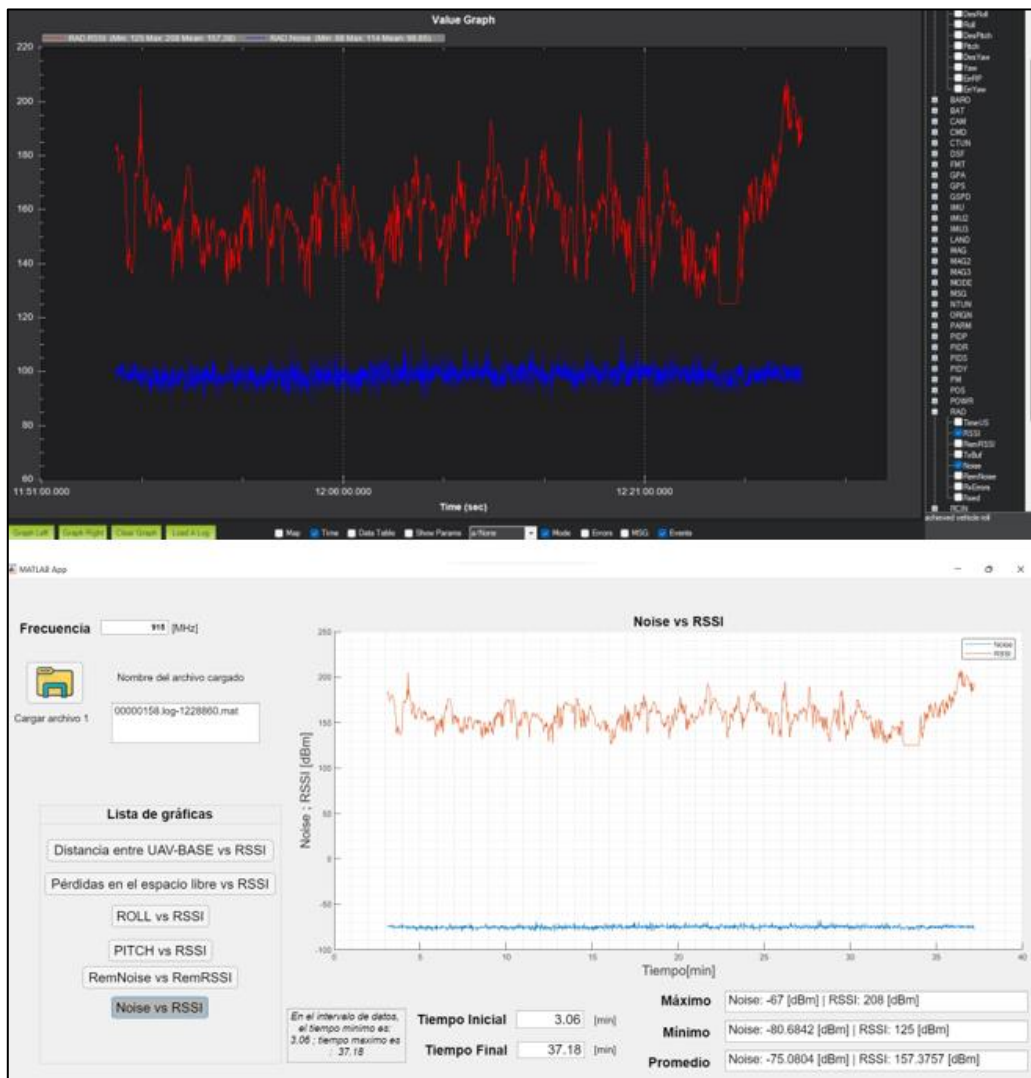


Figura 3.18. Noise vs RSSI | Mission Planner (superior) , Aplicación (inferior)

3.1.3 ANÁLISIS DE PRUEBAS CON EL MISMO RECORRIDO

Para finalizar esta sección se pone a prueba la aplicación con dos archivos que registran planes de vuelo realizados el 6 de febrero del 2023, con los cuales se dispone a realizar un análisis de las gráficas de distancia entre UAV-BASE vs RSSI. Cabe mencionar que los archivos fueron entregados en formato (.bin), a los cuales de inicio se realizó la conversión a formato (.mat) para posteriormente ser cargados en la aplicación. El proceso de conversión de formato se detalla al inicio del presente capítulo. A continuación, se presenta la descripción de los planes de vuelo.

3.1.3.1 Descripción de archivos

- Archivo 2023-02-06 06-41-10.bin

Tal y como se puede observar en la figura 3.19, el plan de vuelo que se sigue va entre 4 puntos principales de tal manera que la trayectoria es de forma ovalada y periódica.



Figura 3.19. Plan de vuelo | Archivo 2023-02-06 06-41-10.bin

Dentro de los cambios con respecto a la antena, se realizó un cambio en la alineación, dividiendo el tiempo de vuelo en dos partes: primera parte donde la antena en el UAV está

en posición horizontal y la antena en la estación terrena también se puso en forma horizontal y segunda parte donde la antena en el UAV está en posición horizontal y la antena en la estación terrena se colocó con una inclinación de 45°.

- Archivo 2023-02-06 07-47-08.bin

Tal y como se puede observar en la figura 3.20, el plan de vuelo que se sigue va entre 4 puntos principales que son exactamente los mismos que el archivo anterior por lo que de igual manera la trayectoria es de forma ovalada y periódica. Notar que en este caso las líneas se encuentran más juntas lo que significa que las vueltas fueron más similares entre sí, en comparación con el plan de vuelo del archivo anterior.

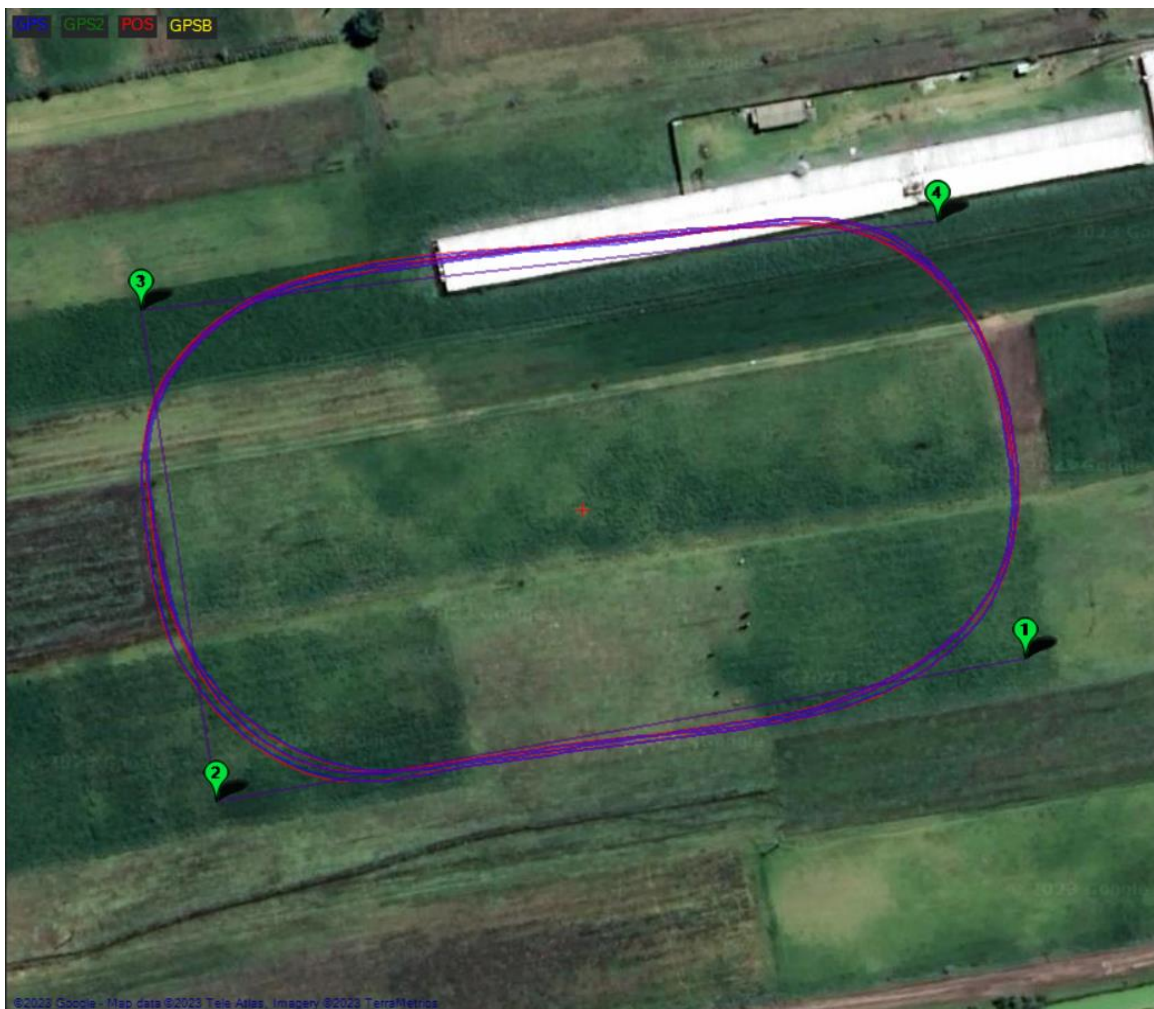


Figura 3.20. Plan de vuelo | Archivo 2023-02-06 07-47-08.bin

Dentro de los cambios con respecto a la antena, se realizó un cambio en la alineación que se mantiene en todo el intervalo, de tal manera que la antena en el UAV y en la estación terrena van en forma vertical. A continuación, se presentan los resultados obtenidos con la aplicación.

3.1.3.2 Análisis de resultados

A continuación, se utiliza la aplicación que procesa dos archivos, de tal manera que inicialmente se utiliza el archivo 2023-02-06 06-41-10.bin, que se compone de dos partes, tal y como se describió en la sección anterior. Por lo que se carga el mismo archivo dos veces con el objetivo de proyectar los dos intervalos de tiempo, tal y como se observa en la figura 3.21, inicialmente se puede distinguir líneas con forma periódica para la distancia, lo que indica claramente que se está siguiendo un movimiento periódico, es decir que el UAV se va acercando y alejando de la estación terrena en varias ocasiones. Siendo así que cuando se tenga mayor distancia se tiene menores valores de RSSI.

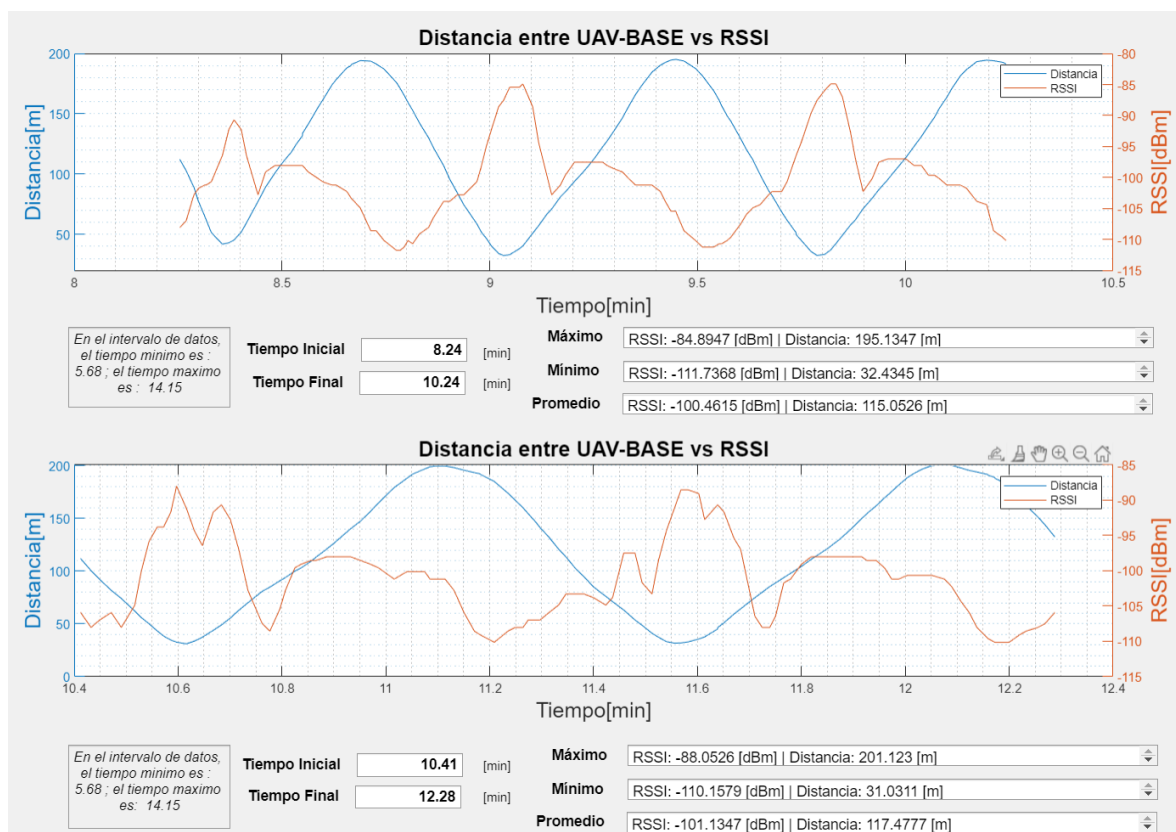


Figura 3.21. Distancia entre UAV-BASE vs RSSI | primera parte (superior) , segunda parte (inferior)

Ahora bien, debido a que en las líneas se tiene un patrón de comportamiento bastante similar, se centra la atención en los valores máximos y promedios donde se puede evidenciar mayores diferencias entre los tramos. Inicialmente se nota que la distancia promedio es bastante similar en ambas partes, de tal manera que se tiene 115.05 [m] y 117.47[m] para la primera y segunda parte respectivamente, lo cual era un resultado esperado debido a que como se observó anteriormente se estaba siguiendo una trayectoria periódica en forma ovalada en todo el plan de vuelo.

Por el contrario, existe un cambio más significativo en los valores de RSSI, donde en la primera parte se obtiene un máximo de -84.89 [dBm] mientras que en la segunda parte se obtiene un máximo de -88.05 [dBm], de tal manera que se puede notar que se tiene más potencia de señal en la primera parte, es decir cuando ambas antenas se encuentran alineadas de manera horizontal, en contraste se presenta menos nivel de potencia de señal cuando se realiza el cambio de la alineación a 45° en la estación base.

A continuación, se presenta el análisis del segundo archivo 2023-02-06 07-47-08.bin , que como se mostró anteriormente, se sigue el mismo plan de vuelo, el único cambio es la alineación de las antenas, que en esta ocasión va de forma vertical y se realiza una comparación con el primer archivo en la parte en la que las antenas van de forma horizontal, tal y como se puede observar en la figura 3.22. Inicialmente, se nota que de igual forma que al análisis anterior se tiene líneas con patrones repetitivos para la distancia.

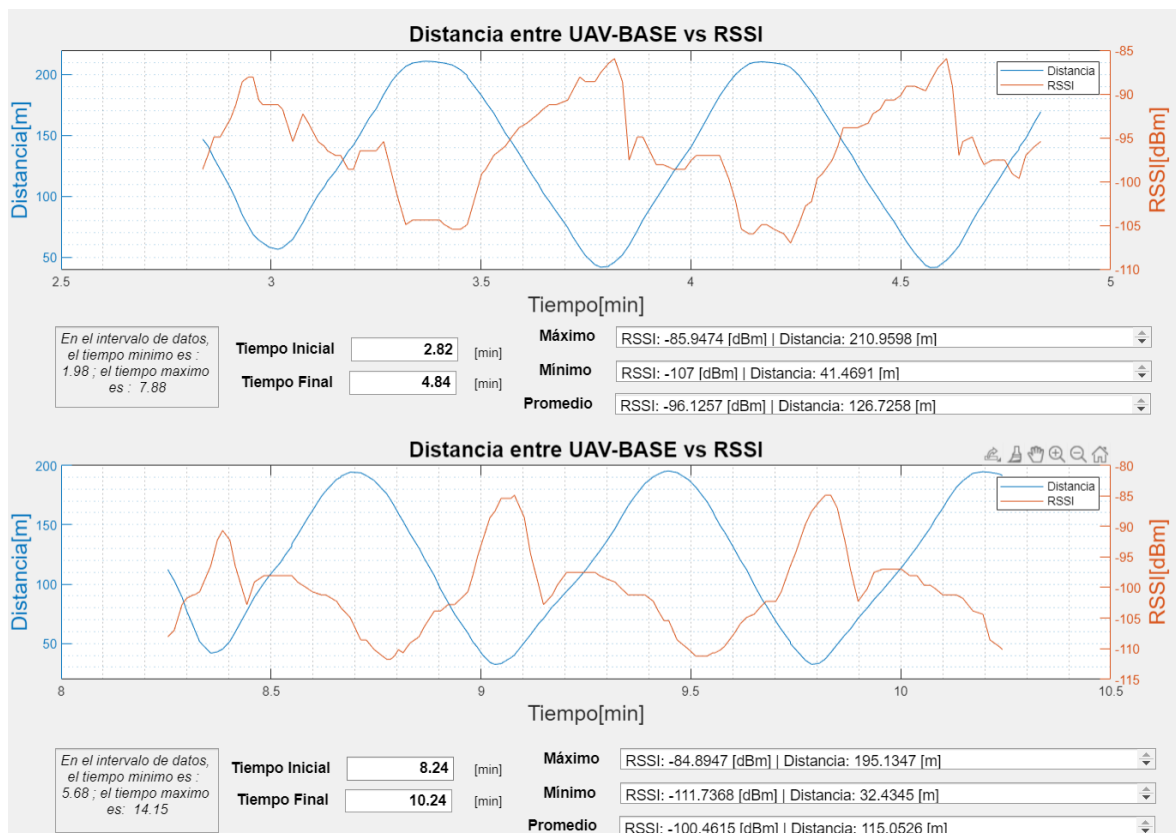


Figura 3.22. Distancia entre UAV-BASE vs RSSI | alineación vertical (superior) , alineación horizontal (inferior)

Ahora comparando los valores de RSSI, se puede observar que en promedio existe una diferencia significativa donde se tiene -96.12 [dBm] para la alineación vertical mientras que se tiene -100.46 [dBm] para la alineación horizontal. Además, se nota que se tiene unos

valores máximos con poca variación, mientras que en los valores mínimos se presentan una diferencia significativa donde se tiene -107 [dBm] para la alineación vertical y -111 [dBm] para la alineación horizontal. Siendo así que se tiene valores más altos de potencia recibida en la configuración de alineación vertical.

De todo esto se puede concluir que para el presente plan de vuelo y realizando cambios en alineación de las antenas, se tiene que la configuración UAV-BASE horizontal-inclinación 45° presenta los peores valores de potencia de la señal recibida. Para la configuración UAV-BASE horizontal-horizontal se tiene valores de potencia intermedios, mientras que la configuración UAV-BASE vertical-vertical presenta los mejores resultados en promedio de tal manera que es la configuración más recomendada para un plan de vuelo de características similares.

3.2 CONCLUSIONES

- La presente interfaz permite graficar datos de la calidad del enlace de telemetría lo cual brinda varios beneficios que pueden mejorar enormemente el proceso de toma de decisiones dentro de los planes de vuelo de los UAV . Algunos de los beneficios clave son: la representación visual de los datos, registro de gráficas, comparación de archivos y selección de intervalos de tiempo .
- Dentro de un archivo log el parámetro más importante para la evaluación de la calidad del enlace es el RSSI, de tal manera que proporciona una indicación de la fuerza de la señal recibida, donde una señal más fuerte, da como resultado una mejor calidad de enlace con menos interferencia y menos errores de transmisión.
- El parámetro de la distancia va de la mano con las pérdidas en el espacio libre de tal manera que se encuentran en una relación directa y ambas en una relación inversa con el RSSI, siendo así que cuando se tenga más distancia, se tendrán más pérdidas y por consiguiente menor fuerza de la señal recibida.
- El parámetro roll y pitch presenta valores para la interpretación de los giros del UAV en el espacio, y por consecuencia de dicho movimiento se espera una desalineación de las antenas que incurre en pérdidas del nivel de señal. En el presente trabajo se concluye que dichas pérdidas no son significativas en comparación con las pérdidas ocasionadas por la distancia.
- Teniendo un mismo plan de vuelo donde se siga una trayectoria periódica, se presentan los peores niveles de señal cuando se tiene un tipo de alineación de la antena del radio de telemetría en un extremo y un diferente tipo de alineación en el otro extremo. Por el contrario, cuando se tiene la misma alineación se pueden obtener mejores resultados en la intensidad de la señal recibida. Ahora bien, la configuración que presentó mejores resultados fue en la que en los ambos extremos se tuvo una alineación vertical.
- Tener una interfaz de procesamiento de dos archivos a la vez , presenta ventajas al momento de querer analizar dos intervalos de tiempo diferentes de un mismo plan de vuelo, de tal manera que se pueda identificar contrastes entre los tramos de interés. Así mismo, se puede realizar la comparación de dos archivos log diferentes pero que en esencia hayan seguido el mismo de plan de vuelo con el objetivo de visualizar correlaciones o diferencias.

3.3 RECOMENDACIONES

- Se recomienda hacer el análisis de los parámetros inicialmente cargando todos los sub-archivos en el Workspace de Matlab con el objetivo de poder visualizar de manera sencilla cada una de las carpetas, tipos y tamaño de datos.
- Antes de iniciar el diseño de la interfaz se recomienda tener en cuenta las herramientas y funciones disponibles en la programación para cumplir con todas las funcionalidades propuestas.
- Se recomienda llevar a cabo pruebas de funcionamiento con varios archivos log, con el objetivo de identificar fallos en el código. Así mismo se recomienda comprobar la aplicación final en varios equipos a fin de detectar fallos entre las versiones de Matlab o el sistema operativo.
- Para trabajos futuros se recomienda aplicar procesamiento de señales, como el suavizado de las líneas por ejemplo para que en las gráficas del RSSI no se presenten tantos picos de variación, sino únicamente los puntos y curvas más destacables.
- Para otro trabajo futuro se recomienda agregar segmentos de código que permitan el análisis de los archivos log, independientemente si está en formato (.bin), (.log) o (.mat) con el objetivo de evitar el uso de Mission Planner.

4. REFERENCIAS BIBLIOGRÁFICAS

- [1] A. Barrientos, J. Cerro, P. Guitiérrez and R. San Martin, "Vehículos aéreos no tripulados para uso civil. Tecnología y aplicaciones". Madrid: Grupo de Robótica y Cibernética, Universidad Politécnica de Madrid, 2022.
- [2] Balaguer, "Análisis y estudio de mejora de un UAV de ala fija VTOL". Universitat Politècnica de Catalunya, 2022.
- [3] M. Moscoso and E. Yáñez, "SISTEMA DE TELEMETRÍA Y VIDEO VIGILANCIA PARA EL CONTROL Y MONITOREO DE MISIONES A LARGA DISTANCIA PARA EL CENTRO DE INVESTIGACIÓN Y DESARROLLO DE LA FAE". Ambato: Universidad Técnica de Ambato, 2022, p. 27.
- [4] I. García, "Estudio sobre vehículos aéreos no tripulados y sus aplicaciones", Universidad de Valladolid. Escuela de Ingenierías Industriales, 2017.
- [5] K. Valavanis and G. Vachtsevanos, "Handbook of unmanned aerial vehicles". Dordrecht: Springer, 2015.
- [6] M. Rodriguez, D. Pigatto, J. Fontes y A. Pinto, "UAV Integration Into IoT: Opportunities and Challenges," de *The Thirteenth International Conference on Autonomic and Autonomous Systems (ICAS)*, 2017.
- [7] D. Cornejo y J. Tintin, "Diseño, construcción e implementación de un sistema de telemetría utilizando tecnología GSM", Universidad Politécnica Salesiana. Facultad de Ingenierías, 2010.
- [8] S. Cayo y I. Changoluisa, "Sistema de seguimiento de un UAV para establecer comunicacion con una estacion de monitoreo en Tierra," 2018. [Online]. Available: <http://bibdigital.epn.edu.ec/handle/15000/19755>. [Accessed: 09-Nov-2022].
- [9] Tomás Brancalenti, Gabriel Flippa, "Transmisión de datos por RF terrestre". Universidad Tecnológica Nacional, 2003.
- [10] C. Yan, L. Fu, J. Zhang, and J. Wang, "A comprehensive survey on UAV Communication Channel Modeling," *IEEE Access*, vol. 7, pp. 107769–107792, 2019.
- [11] RotorDroneMag.com, "Drone FPV Antennas," RotorDrone, [Online]. Available: <https://www.rotordronepro.com/guide-fpv-antennas/>. [Accessed: 09-Nov-2022].
- [12] "UAV antenna MP900," MP Antenna, 17-Jan-2023. [Online]. Available: <https://www.mpantenna.com/product/uav-antenna-mpsl900/>. [Accessed: 5-Jan-2023].
- [13] "900 mhz 3DBI base antenna," Harris Aerial, 24-May-2017. [Online]. Available: <https://www.harrisaerial.com/product/900-mhz-3dbi-base-antenna/>. [Accessed: 5-Jan-2023].

- [14] "Antena 900 mhz 2.1 dBi Monopolowa (RPSMA) - SKLEP BZB UAS," BZB UAS, 28-Sep-2022. [Online]. Available: <https://bzbuas.com/sklep/komponenty-elektroniczne/lacznosc/anteny/antena-900-mhz-2-1-dbi-monopolowa-rpsma/>. [Accessed: 5-Jan-2023].
- [15] "FPV drone antenna mounting guide," *GetFPV Learn*, 17-Jan-2019. [Online]. Available: <https://www.getfpv.com/learn/fpv-essentials/fpv-drone-antenna-mounting-guide/>. [Accessed: 09-Nov-2022].

5. ANEXOS

ANEXO I. Manual de usuario para instalación de la aplicación

ANEXO II. Manual de usuario para manejo de la aplicación

ANEXO III. Anexo digital de secciones de código archivos y funciones de la aplicación

ANEXO I

De manera inicial se entregará al usuario la aplicación que se puede observar en la figura 5.1, el cual es un instalador que permite la descarga de los paquetes necesarios para el completo funcionamiento de la aplicación.

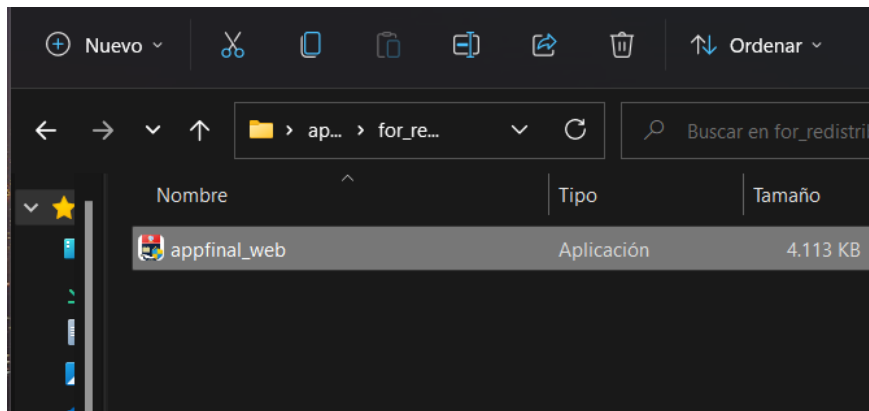


Figura 5.1. Instalador de la aplicación final

De manera seguida se despliega la ventana de la figura 5.2, en la cual se presenta la barra de avance en la preparación de la instalación. En este punto cabe mencionar que es recomendable tener una buena conexión a internet.

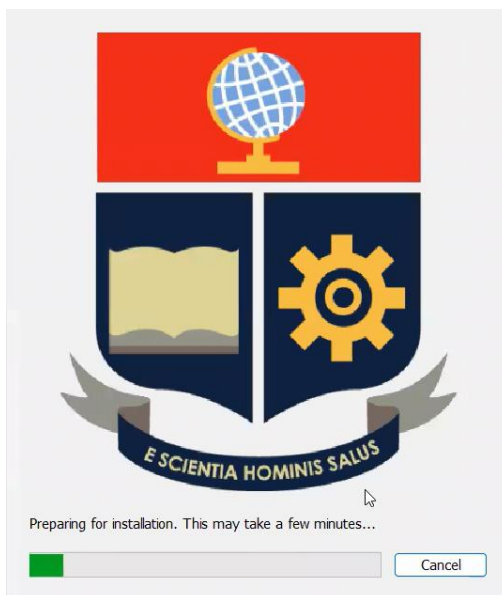


Figura 5.2. Ventana de preparación de instalación

A continuación, se despliega la ventana de inicio de instalación, tal y como se observa en la figura 5.3 donde es necesario dar clic en la opción “next” para continuar.



Figura 5.3. Ventana de inicio de instalación

Continuando con la instalación, se despliega la ventana para la descarga del software “MATLAB Runtime”, el cual sirve como compilador de la aplicación, teniendo en cuenta que todas las interfaces fueron desarrolladas dentro del entorno de Matlab.

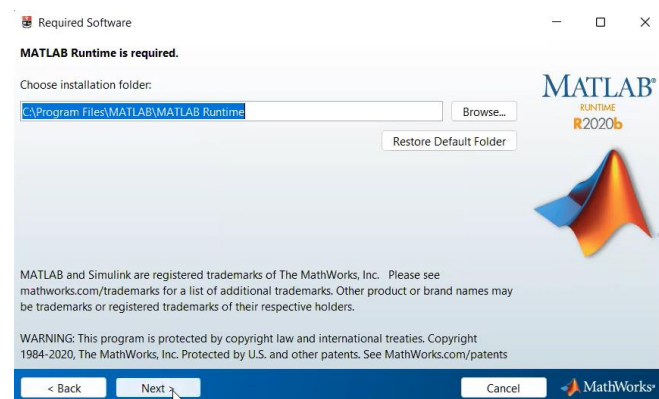


Figura 5.4. Ventana de instalación de MATLAB Runtime

A continuación, es necesario aceptar los términos y condiciones de la descarga, tal y como se presenta en la figura 5.5.

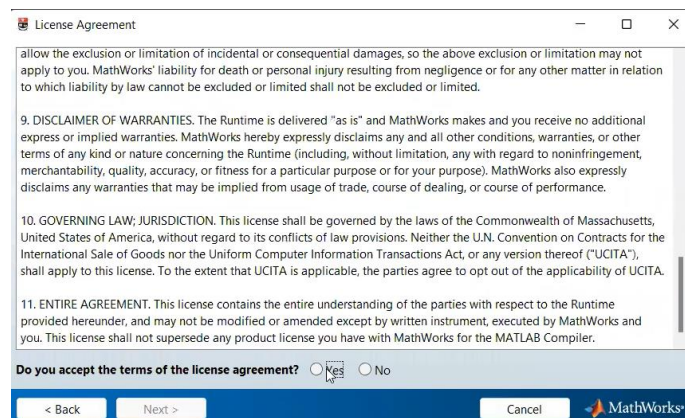


Figura 5.5. Ventana de aceptación de términos y condiciones

De manera seguida se despliega la siguiente ventana de la figura 5.6, en la que se presenta un resumen de los elementos a descargar e instalar. Notar que es necesario 722 MB de espacio disponible en la PC. Finalmente dar clic en “Install”.

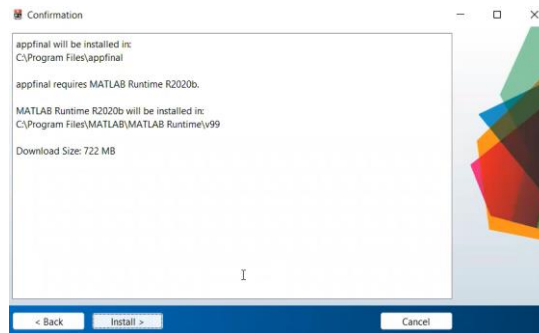


Figura 5.6. Ventana de confirmación de instalación

Una vez que se inicia la instalación, se procede a la descarga de todos los archivos tal y como se observa en la figura 5.7.

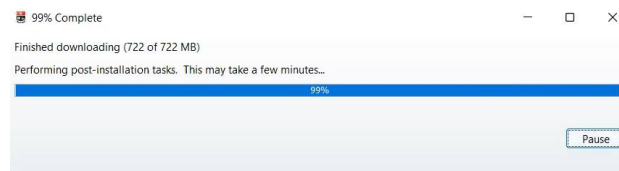


Figura 5.7. Barra de avance de descarga de paquetes

Al finalizar la instalación el programa quedará guardado e instalado dentro de Windows, de tal manera que es necesario buscarlo dentro de las aplicaciones, tal y como se observa en la figura 5.8.

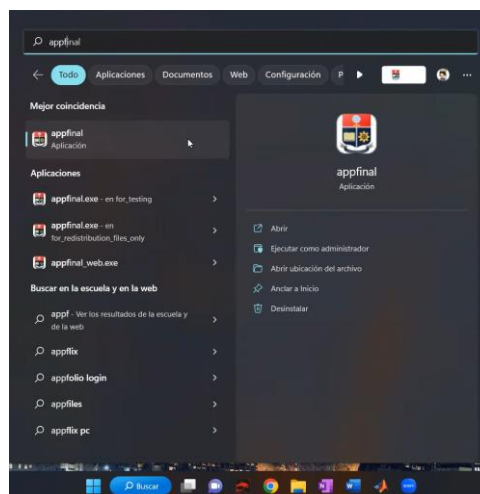


Figura 5.8. Aplicación instalada dentro de Windows

Llegando a este punto la aplicación se encuentra instalada y lista para su funcionamiento.

ANEXO II

Manual de usuario para manejo de la aplicación

De manera inicial se despliega la ventana de presentación, tal y como se presenta en la figura 5.9.



Figura 5.9. Ventana de presentación

A continuación, se incluye el funcionamiento de los botones incluidos dentro de la ventana.

1. Botón para selección de interfaz para procesamiento de un solo archivo.
2. Botón para selección de interfaz para procesamiento de dos archivos.

Una vez seleccionado algún botón automáticamente se va a cerrar la ventana de presentación y de manera seguida se abre la ventana de la opción seleccionada.

La ventana para procesamiento de un solo archivo se muestra en la figura 5.10

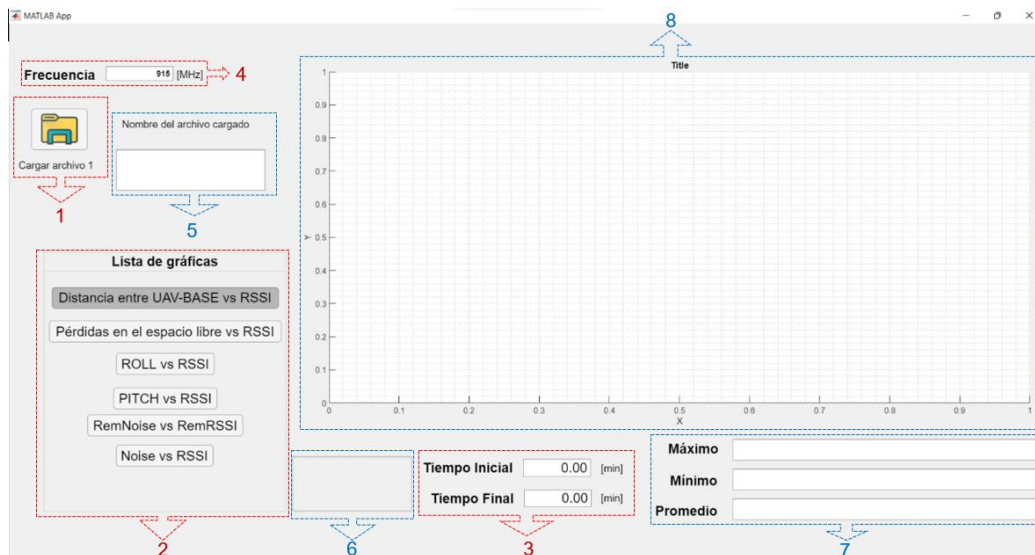


Figura 5.10. Ventana para procesamiento de un archivo

A continuación, se presenta el funcionamiento de los botones incluidos dentro de la ventana.

1. Botón para carga de archivo

Dar clic en el botón “Cargar archivo 1”, para la selección del archivo log dentro del explorador de Windows. Hay que recordar que el archivo debe encontrarse en formato (.mat).

De manera automática se despliega el nombre del archivo en la caja de información “Nombre de archivo cargado” y se despliega la primera gráfica “Distancia entre UAV-BASE vs RSSI” con toda la información de tiempo, datos de los valores máximos, mínimos y promedios.

Si se desea cargar un nuevo archivo, únicamente volver a dar clic en el botón.

2. Panel de botones para despliegue de gráficas

Una vez cargado un archivo, dar clic en el título de la gráfica que se desee visualizar y automáticamente se va a desplegar, en conjunto con la información de valores máximos, mínimos y promedios.

3. Caja editable de tiempo inicial y final

Una vez seleccionada y desplegada la gráfica. Se puede escoger entre un intervalo de tiempo. Para ello se debe dar clic para la edición del tiempo inicial y tiempo final. Hay que recordar que este valor debe ser en minutos y puede ser entero o con decimales separado por un punto. En este punto cabe mencionar que, aunque se puede insertar gran cantidad de decimales, se realiza un redondeo para admitir únicamente dos.

Una vez realizado el cambio, automáticamente se desplegará la gráfica en dicho intervalo de tiempo. Así mismo se muestra los valores de máximo, mínimo y promedio dentro del nuevo intervalo.

Finalmente, se conserva el intervalo de tiempo aun cuando se vaya a cambiar de gráfica. Para volver a presentar la gráfica completa es necesario configurar al intervalo de inicio, para ello se puede observar en la caja 6, donde se tiene la información del tiempo máximo y mínimo que se tiene en el archivo.

4. Caja editable de frecuencia

Dar clic en la caja para la edición del valor de frecuencia. Tener en cuenta que el valor admitido se encuentra en MHz, el cual puede ser entero o con decimales separado por un punto. Además, hay que recordar que una vez editada la frecuencia, los cambios se proyectaran únicamente en la gráfica "Pérdidas en el espacio libre vs RSSI" debido a que es la única que depende de la frecuencia, así mismo se notarán cambios en los valores de máximo, mínimo y promedio únicamente del parámetro de pérdidas.

5. Caja con el nombre del archivo cargado actualmente

En esta sección se presenta el nombre del archivo que se encuentra cargado actualmente, el cual se va a ir actualizando en cuanto se cargue un nuevo archivo.

6. Caja con el intervalo de tiempo del que se tiene información

En esta sección se presenta el tiempo mínimo y máximo del que se tiene información en el archivo que se encuentra cargado actualmente. Esta información puede ser útil cuando se haya hecho una configuración al intervalo de tiempo de las gráficas y se desee volver a desplegar la gráfica completa. Esta información se va a ir actualizando en cuanto se cargue un nuevo archivo.

7. Caja con datos mínimos, máximos y promedios de la gráfica seleccionada

En esta sección se presenta el valor máximo, mínimo y promedio de los parámetros que intervienen en la gráfica seleccionada al momento. Así mismo se presentan dichos valores en función del intervalo de tiempo seleccionado en el momento.

8. Eje de gráficas

En esta sección se muestran las gráficas seleccionadas al momento, con su respectivo título, leyendas, nombres de los ejes y unidades de medida. Inicialmente se presenta la gráfica completa, pero se va actualizando en función de que exista algún cambio en el intervalo de tiempo.

Si se desea finalizar el uso de la ventana para el procesamiento de un archivo únicamente se debe cerrar y de manera automática se desplegará la ventana de presentación.

La ventana para procesamiento de dos archivos se muestra en la figura 5.11

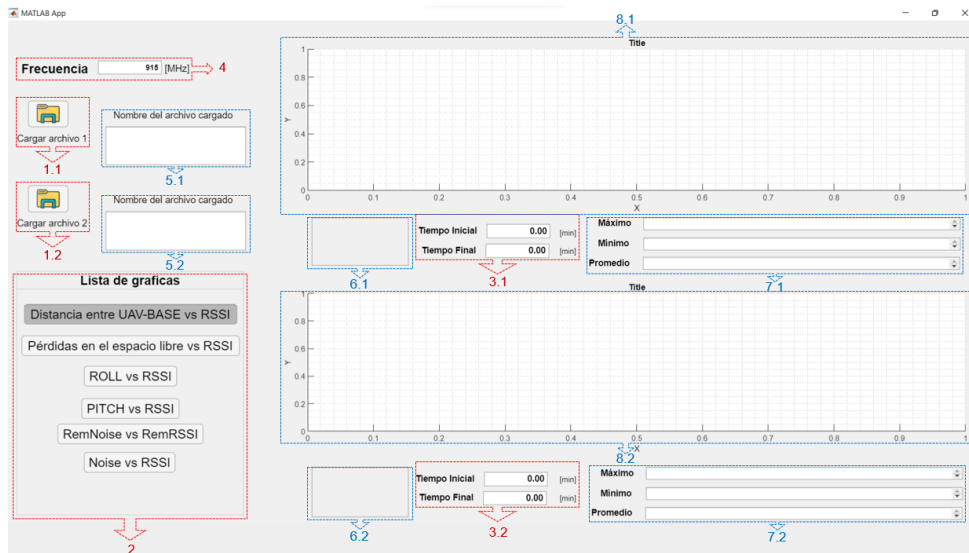


Figura 5.11. Ventana para procesamiento de dos archivos

A continuación, se presenta la lista de los botones de la ventana de la figura 5.11

- 1.1 Botón para carga de archivo 1.
- 1.2 Botón para carga de archivo 2.
2. Panel de botones para despliegue de gráficas.
- 3.1 Caja editable de tiempo inicial y final de archivo 1.
- 3.2 Caja editable de tiempo inicial y final de archivo 2.
4. Caja editable de frecuencia.
- 5.1 Caja del nombre del archivo 1 cargado actualmente.
- 5.2 Caja del nombre del archivo 2 cargado actualmente.
- 6.1 Caja con el intervalo de tiempo del que se tiene información del archivo 1.
- 6.2 Caja con el intervalo de tiempo del que se tiene información del archivo 2.
- 7.1 Caja con datos mínimos máximos y promedios de la gráfica seleccionada del archivo 1.
- 7.2 Caja con datos mínimos máximos y promedios de la gráfica seleccionada del archivo 2.
- 8.1 Eje de gráficas del archivo 1.
- 8.2 Eje de gráficas del archivo 2.

Como se puede observar se tiene los mismos botones y cajas de información que en la interfaz para procesamiento de un archivo, de tal manera que su funcionamiento es completamente el mismo.

En este punto cabe mencionar que se deja intencionalmente un solo panel de gráficas con el objetivo de que se tenga seleccionado el mismo título de gráfica para los dos archivos de tal manera que se puede realizar un análisis entre los mismos parámetros.