



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

MODELACIÓN ESTADÍSTICA DE CANAL SATELITAL PARA APLICACIÓN EN
MICROSATÉLITES

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELECTRICISTA

CAMILO JOSÉ ROJAS MILLA

SANTIAGO DE CHILE
AGOSTO 2012



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

MODELACIÓN ESTADÍSTICA DE CANAL SATELITAL PARA APLICACIÓN EN
MICROSATÉLITES

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELECTRICISTA

CAMILO JOSÉ ROJAS MILLA

PROFESOR GUÍA:
MARCOS DÍAZ QUEZADA

MIEMBROS DE LA COMISIÓN:
PATRICIO PARADA SALGADO
ALEX BECERRA SAAVEDRA

SANTIAGO DE CHILE
AGOSTO 2012

RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELECTRICISTA
POR: CAMILO JOSÉ ROJAS MILLA
FECHA: AGOSTO 2012
PROF. GUÍA: MARCOS DÍAZ QUEZADA

MODELACIÓN ESTADÍSTICA DE CANAL SATELITAL PARA APLICACIÓN EN MICROSATÉLITES

La explosiva evolución de los sistemas de telecomunicaciones plantea una necesidad por construir esquemas con cada vez mejor rendimiento en tasa de datos y de errores. Un posible acercamiento para mejorar la tasa de error en los esquemas de comunicaciones es la adaptación de la señal en relación a las características del canal por el que viaja. Para realizar esta adaptación es necesario conocer la información instantánea del canal o sus propiedades estadísticas.

En el presente trabajo de memoria se presenta un algoritmo de estimación lineal y otro algoritmo de estimación lineal secuencial capaces de caracterizar la información instantánea de un canal. Para la implementación y el desarrollo de los algoritmos de estimación se utiliza un modelo matemático en banda base discreto del canal (canal TDL) y se realizan suposiciones sobre el tiempo de coherencia y rango del modelo.

Se desarrolló, a través del software Matlab y el toolbox de comunicaciones, una simulación para un canal real inalámbrico y se ajustaron algunos parámetros (como la tasa de datos, esquema de modulación, dispersión de Doppler) de la simulación de acuerdo al esquema de comunicaciones del proyecto SUCHAI, proyecto que tiene como objetivo diseñar y construir un satélite tipo cubesat y lanzarlo en órbita. Frente a estas simulaciones se hicieron múltiples realizaciones para medir el desempeño instantáneo y estadístico de los estimadores.

Se desarrollaron métricas para evaluar el comportamiento de cada estimador. El rendimiento se evaluó obteniendo las características estadísticas del estimador: como la varianza y el error cuadrático medio entre la media del estimador y el valor real de la simulación; y las características estadísticas del canal estimado: como la correlación de estados y los parámetros K y Ω de la distribución Riciana.

Se comparó el rendimiento de ambos estimadores, llegando a la conclusión que ambos son capaces de estimar un canal con las cualidades simuladas, sin embargo el estimador secuencial presenta ventajas en rendimiento e implementación por hardware. A través de las simulaciones se definieron límites, de tiempo de coherencia y relación señal a ruido, para los cuales cada estimador produce un buen resultado.

Finalmente, se desarrollaron funciones de estimación directa (lineal y secuencial) capaces de caracterizar un canal en función de las señales enviada y recibida, junto con algunos parámetros del canal como el tiempo de coherencia, rango del modelo y frecuencia de muestreo. Se desarrolló, también, una metodología para obtener los parámetros necesarios para las funciones.



... para mi familia y amigos.

Agradecimientos

Agradezco a mi familia por todo el apoyo brindado en estos largos años. Principalmente agradezco a mi madre y padre por la ayuda emocional y económica, sin la cual obtener el título hubiera sido imposible. También agradezco a Paulina por ayudarme a imprimir la versión final de este documento.

Agradezco, también, a toda mi comisión; al profesor Marcos por su ayuda, buena onda y disposición; al profesor Patricio por su conocimiento infinito sobre el tema, y buena disposición a compartirlo; y finalmente al Alex por sus buenas ideas, experiencia y soporte.

Muchas gracias a todos los integrantes del proyecto SUCHAI y en especial a mi compañeros de equipo, Sebastián y Felipe por su apoyo y ayuda en el proceso tortuoso que fue realizar esta memoria.

Finalmente, gracias a todos mis amigos por soportarme y ayudarme a soportar el estrés en espacios fuera de lo académico. Trataré de listar algunos pero probablemente se me olviden unos cuantos. Gracias a Nicolás, Chino, Velin, Maryon, Naty, Juan, Gabriel, Rodrigo, Sebastián (Riffo), Pablo, Moisés, Joaco y a todos los que, de alguna u otra forma, me ayudaron a mantener la sanidad mental durante mi estadía en la facultad.

Índice general

1. Introducción	9
1.1. Objetivos	10
1.1.1. Objetivo general	10
1.1.2. Objetivos Específicos	10
1.2. Estructura del informe	11
2. Antecedentes	12
2.1. Esquema de comunicaciones digitales	12
2.1.1. Canal de comunicaciones	13
2.1.2. Modulación digital	16
2.2. Procesamiento Digital de Señales	18
2.2.1. Señales	18
2.2.2. Teorema del muestreo	19
2.2.3. Conversión de señal analógica a digital	20
2.3. Comunicaciones Inalámbricas	20
2.3.1. Ondas electromagnéticas	21
2.3.2. Ondas electromagnéticas reflejadas	22
2.3.3. Modelo para comunicaciones inalámbricas	23
2.3.4. Modelo discreto para comunicaciones inalámbricas	24
2.3.5. Ruido blanco aditivo Gaussiano	25

2.3.6.	Tiempo de Coherencia	25
2.3.7.	Modelos estadísticos de canales	26
2.4.	Comunicaciones satelitales	28
2.4.1.	Tipos de órbita	29
2.4.2.	Satélites miniaturizados	31
2.4.3.	Satélites tipo “cubesat”	32
2.4.4.	Proyecto SUCHAI	32
2.5.	Estimación estadística de señales	34
2.5.1.	Calidad del estimador	34
2.5.2.	Cota inferior de Cramer-Rao	35
2.5.3.	Modelos Lineales	36
2.5.4.	Estimador de máxima verosimilitud	37
2.5.5.	Cálculo secuencial del estimador lineal	38
2.6.	Información del estado del canal	40
2.6.1.	Método de secuencias piloto	40
2.7.	Estado del arte	41
3.	Implementación	42
3.1.	Modelo del canal	42
3.2.	Estimadores Implementados	43
3.2.1.	Estimador lineal	43
3.2.2.	Estimador lineal secuencial	45
3.3.	Simulación del canal físico	47
3.3.1.	Comportamiento de la simulación	53
3.4.	Procedimiento de simulación	54
3.4.1.	Procedimiento para obtener las características del estimador	54
3.4.2.	Procedimiento para obtener las características del canal	55

3.5. Secuencias piloto	56
4. Discusión de resultados	57
4.1. Resultados estimador lineal	57
4.1.1. Comportamiento del estimador	57
4.1.2. Comportamiento del canal estimado	61
4.1.3. Secuencias piloto	63
4.2. Resultados estimador secuencial	64
4.2.1. Comportamiento del estimador	64
4.2.2. Comportamiento del canal estimado	68
4.2.3. Secuencias Piloto	70
4.3. Comparación de estimadores	71
4.4. Implementación a canal real	72
5. Conclusiones	75
5.1. Trabajos Futuros	76
Glosario	78
Bibliografía	80
A. Código fuente	I
A.1. Funciones de estimación	I
A.2. Simulaciones	IV

Índice de figuras

2.1. Esquema de comunicación digital	12
2.2. Principales efectos electromagnéticos que pueden afectar un sistema de comunicaciones.	14
2.3. Interferencia constructiva y destructiva	15
2.4. Esquema Modulación FSK	17
2.5. Curva de desempeño teórica de una modulación FSK no coherente utilizando un canal AWG	18
2.6. Esquema de un Sistema	19
2.7. Diagrama de bloques conversor análogo-digital	20
2.8. Propagación electromagnética entre dos puntos fijos.	21
2.9. Propagación electromagnética entre emisor fijo y receptor en movimiento.	22
2.10. Propagación electromagnética con receptor móvil y múltiples caminos	23
2.11. Esquema de comunicación por satélite	29
2.12. Esquema de comunicaciones del proyecto SUCHAI. En la estación terrena; el bloque PC corresponde a un servidor que realiza el almacenamiento y manejo de datos, TNC realiza la comunicación binaria entre el PC y el transceiver que transmite y recibe la señal análoga a través de la antena terrestre. En la parte satélital se recibe y envía la señal análoga a través de la antena y transceiver.	33
2.13. Secuencias piloto	41
3.1. Esquema del canal TDL	42
3.2. Esquema del canal completo	43

3.3.	Algoritmo de estimación lineal	45
3.4.	Diagrama de bloques que representa el algoritmo secuencial	46
3.5.	Algoritmo de estimación secuencial lineal	47
3.6.	Esquema de la simulación del canal	47
3.7.	Mensaje aleatorio de 30 bits	48
3.8.	Histograma de $ h_0[n] $. La simulación se realizó utilizando el canal Riciano de matlab con los parámetros mencionados en la tabla 3.3. Se procesaron 16.000 muestras aleatorias, obteniéndose un valor central de 1,0080 y desviación estándar de 0,2130. La simulación se encuentra en el script rician.mat	51
3.9.	Comportamiento del modulador. La línea continua representa el BER teórico de una modulación FSK no coherente binaria, la línea segmentada son los valores medios obtenidos por los bloques fkmod() y fskdemod() para diferentes valores de E_b/N_0 producidos por el bloque awgn() y la línea de tendencia es la función exponencial que mejor se ajusta a los datos. La simulación se hizo con la herramienta “bertool” de Matlab que realiza un test de montecarlo hasta llegar a 100 errores o 10^8 bits para cada E_b/N_0 . el código de la simulación se encuentra en el archivo fksim.m	53
3.10.	Comportamiento de la simulación con canal Riciano y ruido AWG. La línea continua representa el comportamiento teórico de un modulador FSK no-coherente, los datos de la simulación se obtuvieron utilizando el método de Montecarlo hasta llegar a 100 errores o procesar 10^8 bits y la línea de tendencia es la función exponencial que mejor se ajusta a los datos. El código de la simulación se encuentra en el archivo riciansim.m	54
3.11.	Procedimiento de simulación para estimación lineal	55
4.1.	Magnitud de $h_0[n]$. La línea punteada es el valor teórico de la magnitud de $h_0[n]$ del canal Riciano fijado para la simulación y la línea continua es el valor calculado a través de la ecuación 4.1 con las estimaciones obtenidas de 100 realizaciones, utilizando el algoritmo mostrado en la sección 3.4.1 con el estimador lineal. El código de la simulación se encuentra en el archivo algoritlin.m	58
4.2.	Magnitud de $h_\ell[n]$, $\ell > 0$. La metodología de obtención de datos es idéntica a la utilizada en la figura 4.1 pero se exponen los datos de $h_1[n]$ y $h_2[n]$	59
4.3.	Varianza del estimador lineal. Donde los parámetros teóricos de cada estimador son calculados a través de la matriz de covarianza obtenida por la ecuación 4.4 y los valores estimados se obtienen a través de la ecuación 4.2. La simulación se realizó implementando el algoritmo de la sección 3.4.1.	59

4.4. Error cuadrático medio entre los valores estimados y teóricos. Cada función representa el MSE, calculado a través de la ecuación 4.5, para cada “tap” ℓ . La simulación se realizó implementando el algoritmo presentado en la sección 3.4.1. El código de la simulación se encuentra en el archivo `algoritlin.m` 61

4.5. Correlación de estados $h_0[n]$. La línea punteada representa la correlación temporal del estado h_0 del canal estimado mientras la línea continua representa la correlación del mismo estado para el canal estimado. La simulación se realizó implementando el algoritmo mostrado en la sección 3.4.2. El código de la simulación se encuentra en el archivo `matrizcorrlin.m` 62

4.6. Estimación de Parámetros K y Ω . Los datos se obtuvieron realizando el algoritmo de simulación presentado en la sección 3.4.2 . Se estimó los parámetros K y Ω a través del paquete “Rician” de Matlab, para cada realización, y se calculó su promedio y desviación standar. 63

4.7. MSE de estado $h_0[n]$ para diferentes secuencias. La simulación se realizó implementando el algoritmo presentado en la sección 3.4.1 pero utilizando múltiples largos de secuencia. El valor MSE se calculó a través de la ecuación 4.5. 64

4.8. Magnitud de $h_0[n]$ para estimador secuencial para $E_b/N_0 = 15[dB]$. La simulación se desarrolló implementando el algoritmo presentado en la sección 3.4.1 utilizando el estimador secuencial. El código de la simulación se encuentra en el archivo `algoritsec.m`. 65

4.9. Magnitud $h_\ell[n]$, $\ell > 0$ para un $E_b/N_0 = 15[dB]$. Se utiliza la misma simulación de la figura 4.8 pero se estiman los otros estados. 66

4.10. Error cuadrático medio entre los valores estimados y teóricos; Cada función representa el MSE, calculado a través de la ecuación 4.5, para cada “tap” ℓ . La simulación se realizó implementando el algoritmo presentado en la sección 3.4.1, utilizando el estimador secuencial. El código de la simulación se encuentra en el archivo `algoritsec.m`. 67

4.11. Varianza del estimador secuencial. Donde los parámetros teóricos y estimados son calculados a través de las ecuaciones 4.9. La simulación se realizó implementando el algoritmo de la sección 3.4.1. 68

4.12. Correlación de estados $h_0[n]$ utilizando el estimador secuencial. La línea punteada representa la correlación temporal del estado h_0 del canal estimado mientras la línea continua representa la correlación del mismo estado para el canal estimado. La simulación se realizó implementando el algoritmo de la mostrado en la sección 3.4.2. El código de la simulación se encuentra en el archivo `matrizcorrsec.m` 69

4.13. Estimación de Parámetros K y Ω utilizando el estimador secuencial. Los datos se obtuvieron realizando el algoritmo de simulación presentado en la sección 3.4.2. Se estimó los parámetros K y Ω a través del paquete “Rician” de Matlab, para cada realización, y se calculó su promedio y desviación standar. 70

4.14. MSE de estado $h_0[n]$ para diferentes secuencias utilizando el estimador secuencial. La simulación se realizó implementando el algoritmo presentado en la sección 3.4.1 pero utilizando múltiples largos de secuencia. El valor MSE se calculó a través de la ecuación 4.5. 71

4.15. Esquema de entradas y salida de funciones hlineal.m y hsec.m 72

4.16. Forma de obtener la entrada **rxSig** para enlace de bajada 74

Índice de tablas

2.1. Características de satélites Geoestacionarios	30
2.2. Características satélites de baja órbita	30
2.3. Características satélites de mediana órbita	31
2.4. Características satélites de órbita altamente elípticas	31
3.1. Parámetros de función randi()	48
3.2. Parámetros de función fskmod()	49
3.3. Parámetros de función ricianchan()	50
3.4. Parámetros de las secuencias piloto	56

Capítulo 1

Introducción

Los sistemas de telecomunicaciones han sufrido una explosiva evolución desde su invención, con el telegrama, hasta los sistemas actuales, como la red interconectada mundial (Internet), sistemas de broadcasting satelital o la red GPS entre otros. Los esquemas actuales de comunicaciones permiten transmitir información de cualquier tipo, a través de grandes distancias en cuestión de segundos, lo que provoca una cierta necesidad de mantenerse informado y “conectado” con todo el globo en todo momento. Por esta razón, la necesidad de redes con mejor rendimiento y desempeño es un problema del día a día.

Dentro de los sistemas de comunicaciones actuales, los sistemas satelitales representan una alternativa atractiva para fines comerciales y de investigación, ya que posibilitan la comunicación entre dos puntos lejanos del planeta de forma directa. Estos sistemas también entregan la posibilidad de desarrollar investigación en el espacio exterior sin la necesidad de encontrarse presencialmente allá.

En los últimos 15 años se han hecho grandes progresos en el desarrollo de microsátélites universitarios y privados. La razón de esto se debe principalmente al estandar “cubesat”, desarrollado por la Universidad Politécnica del Estado de California y la Universidad de Stanford, que define ciertas características específicas para el desarrollo y construcción de microsátélites, y facilitan el trabajo de las entidades que quieren hacer desarrollo espacial. El estandar “cubesat” abre muchas posibilidades de investigación espacial a precios accesibles para universidades y empresas privadas, lo cual crea un nuevo mercado de desarrollo espacial con precios reducidos.

En los últimos 3 años se ha desarrollado un proyecto, dentro de la Universidad de Chile, con el objetivo de realizar un microsátélite tipo cubesat diseñado y fabricado por alumnos de la misma institución. La realización de este proyecto abre una nueva gama de posibilidades para investigación y desarrollo dentro de las disciplinas que engloba el satélite, y en particular, el área de comunicaciones del satélite representa un nuevo nicho interesante en donde realizar aportes teóricos y técnicos.

Dentro de cualquier esquema de comunicaciones se desea obtener la mejor relación entre la velocidad de datos y la tasa de error del sistema, teniendo siempre en cuenta la complejidad del esquema a utilizar. En comunicaciones inalámbricas una forma de mejorar el rendimiento del esquema de comunicaciones, en prácticamente cualquier métrica de desempeño, consiste en la adaptación de la señal con respecto al canal de comunicaciones [19]. Para realizar la adaptación de señal con respecto al canal es necesario tener conocimiento de la información instantánea o estadística del mismo, que a su vez, debe ser estimada en función de la entrada y salida del canal.

La caracterización de parámetros del canal se realiza sobre un modelo matemático que lo representa. Las cualidades y características éste deben escogerse de acuerdo a la características reales del canal. La caracterización del modelo escogido se realiza utilizando estimadores estadísticos aplicados al procesamiento de señales. La elección e implementación de los estimadores debe realizarse de acuerdo a supuestos que se tengan sobre el comportamiento del canal.

El trabajo realizado consiste en una investigación sobre el desempeño y calidad de diferentes estimadores aplicados al área procesamiento de señales, y en particular, al sistema de comunicación desarrollado para el proyecto SUCHAI. Se realizó un marco teórico con el cual se desarrollaron mecanismos para simular y modelar el canal de comunicaciones del proyecto, luego se analizó el desempeño de los estimadores escogidos en relación a las simulaciones creadas.

La aplicación del trabajo va directamente relacionado con el esquema de comunicaciones del proyecto SUCHAI. Sin embargo, los algoritmos de estimación y modelos de canal propuestos pueden ser utilizados para multitud de canales inalámbricos con características similares a las propuestas. Cambiando ciertos parámetros de simulación y modulación digital los algoritmos desarrollados pueden adaptarse a prácticamente cualquier esquema de comunicación. Sin embargo el desempeño de los estimadores en otras aplicaciones no está garantizado.

1.1. Objetivos

1.1.1. Objetivo general

El objetivo general de este trabajo es desarrollar y analizar algoritmos de estimación capaces de caracterizar el canal de comunicaciones satélite-estación terrena existente en el proyecto SUCHAI.

1.1.2. Objetivos Específicos

- Realizar investigaciones sobre estimadores usados en la actualidad y su aplicación en el ámbito de los microsatelites.

- Desarrollar simulaciones computacionales que emulen el comportamiento del canal presente en el proyecto SUCHAI.
- Desarrollar y postular algoritmos capaces de estimar el canal de la simulación.
- Analizar el rendimiento de los algoritmos desarrollados.
- Comparar el rendimiento obtenido por los estimadores.

1.2. Estructura del informe

La estructura utilizada en este documento para exponer el trabajo realizado es la siguiente:

- **Capítulo 1. Introducción:**
Corresponde a la descripción del tema, la motivación de éste y los alcances y objetivos del trabajo realizado.
- **Capítulo 2. Antecedentes:**
Este capítulo presenta la revisión bibliográfica realizada. Contiene información sobre comunicaciones inalámbricas, modelos inalámbricos, procesamiento digital, esquemas de comunicaciones, comunicaciones satelitales y estimación estadística de señales.
- **Capítulo 3. Implementación:**
Se presenta el trabajo realizado y las herramientas utilizadas para llevarlo a cabo. Se muestra la simulación desarrollada para medir el rendimiento de los algoritmos, el modelo escogido para el canal inalámbrico, los estimadores utilizados junto con su implementación para caracterizar el modelo escogido y los algoritmos de simulación y estimación utilizados para medir el rendimiento de cada estimador.
- **Capítulo 4. Discusión de resultados:**
Se discute los resultados obtenidos por los algoritmos de estimación presentados en la implementación. Se analiza y compara la estimación de estados del canal y su comportamiento estadístico. También se desarrolla una metodología de caracterización real, utilizando los estimadores propuestos.
- **Capítulo 5. Conclusiones:**
Se enumeran las conclusiones del trabajo realizado y se realizan propuestas de posibles trabajos futuros.

Capítulo 2

Antecedentes

2.1. Esquema de comunicaciones digitales

Las comunicaciones digitales tienen como objetivo transmitir un mensaje, en forma de secuencias de bits, desde un emisor hasta un receptor a través de un canal de comunicaciones. La figura 2.1 muestra los elementos que componen un sistema de comunicaciones digital.

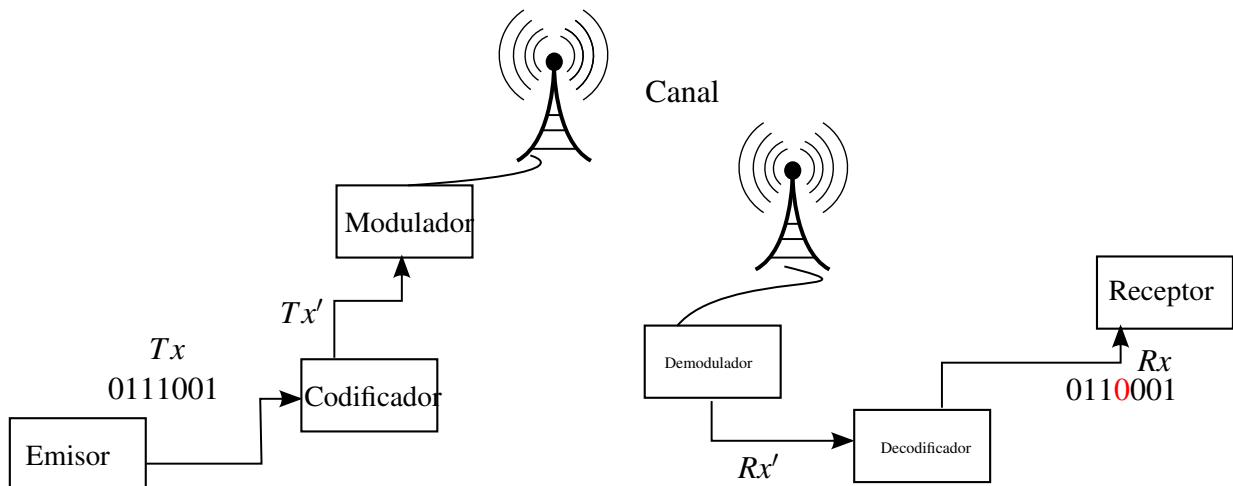


Figura 2.1: Esquema de comunicación digital

El sistema de comunicaciones está compuesto por los siguientes módulos:

1. **Emisor:** Es el que se encarga de enviar la secuencia de bits que contienen la información que se desea comunicar al receptor.

2. **Codificación:** En esta etapa se altera la secuencia de bits original cambiando el orden de estos o agregando bits de redundancia. El esquema de codificación es diseñado para agregar robustez al mensaje enviado y debe ser diseñado en base al modelo del canal.
3. **Modulador Digital:** En esta etapa se desarrolla la conversión entre el mensaje digital y la señal análoga que se transmite por el canal de comunicaciones.
4. **Canal de Comunicaciones:** Es el medio físico por donde viajan las ondas electromagnéticas moduladas en el proceso anterior.
5. **Demodulador Digital:** Recibe las ondas electromagnéticas que viajan por el canal y se demodulan de forma de obtener un mensaje binario.
6. **Decodificador:** Los datos obtenidos del demodulador deben ser decodificados usando el esquema inverso que se usó para codificar. Producto de las etapas anteriores el mensaje obtenido luego de decodificar no siempre será igual al mensaje enviado en un principio.
7. **Receptor:** El receptor obtiene los bits decodificados y los interpreta según se requiera.

Se explicará con mayor profundidad las etapas de modulación digital y canal de comunicaciones.

2.1.1. Canal de comunicaciones

Entre el emisor y el receptor se encuentra el canal a través del cual viajan las señales. El canal representa el medio físico del sistema, que puede ser desde un cable coaxial hasta el vacío. Independiente del medio que se tenga, la mayoría de los canales presentan alteraciones que producen que la transmisión no sea perfecta y que el mensaje recibido varíe con respecto al mensaje original.

Existen tres mecanismos básicos de propagación que impactan en la calidad de transmisión; estos son reflexión, difracción y scattering.

Reflexión

La reflexión es el cambio de dirección del frente de onda producto de un choque con algún obstáculo de grandes dimensiones (en comparación con la longitud de onda de la señal) o de un cambio en el medio de propagación. Producto de este fenómeno, las ondas reflejadas regresan al medio por el que viajaban inicialmente. Las ondas reflejadas pueden interferir constructiva o destructivamente y se producen producto de cualquier objeto que se encuentre en el camino entre el emisor y el receptor.

Difracción

Es un fenómeno que se produce cuando la onda se encuentra con pequeños orificios (en relación a la longitud de onda de la señal), rendijas o bordes. Producto de este fenómeno se producen múltiples ondas que se propagan en todas direcciones con fases diferentes.

Scattering

Scattering o dispersión es el proceso físico en donde la onda se ve obligada a cambiar de dirección producto de un medio no uniforme. Cuando existe solo una deformación que produce una onda se le llama “scattering singular”. En el caso de múltiples deformaciones en el medio que producen múltiples ondas de “scattering”, se le llama “scattering multiple” y es muy complejo predecir las ondas resultantes exactas.

Los fenómenos de reflexión, difracción y scattering se ilustran en la figura 2.2

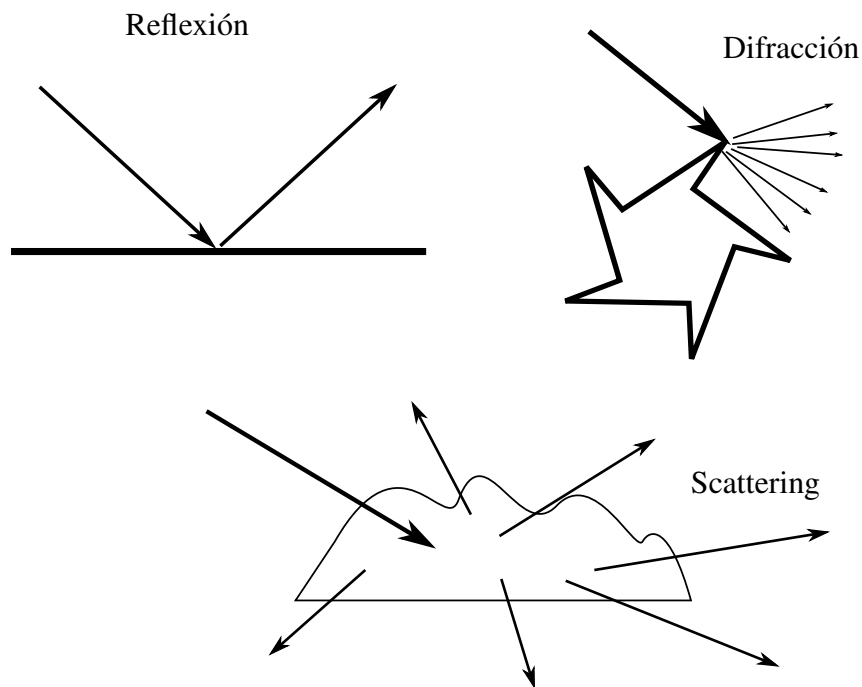


Figura 2.2: Principales efectos electromagnéticos que pueden afectar un sistema de comunicaciones.

Interferencia

Es un proceso físico que ocurre cuando dos o más ondas existen en un mismo punto en el espacio. Debido a lo anterior, se produce una onda resultante que es la suma de todas las ondas que intervienen en el fenómeno. Analizando el caso de dos ondas planas que interfieren, se observan dos tipos de interferencias: la constructiva y la destructiva. Cuando la onda resultante tiene mayor amplitud que las ondas que la conforman, se dice que se produjo una interferencia constructiva y que las ondas están en fase. Por otra parte, si la amplitud de la onda resultante es menor a las ondas que la conforman se produce interferencia destructiva y las ondas están en desfase. Un ejemplo de interferencia constructiva y destructiva se muestra en la figura 2.3.

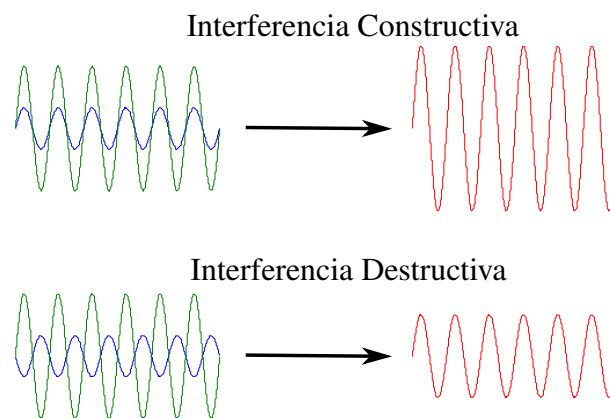


Figura 2.3: Interferencia constructiva y destructiva

Efecto Doppler

El efecto Doppler es el nombre que recibe el fenómeno físico en el cual la frecuencia de la señal percibida por un observador en movimiento relativo a la fuente se ve alterada. Cuando las velocidades del receptor y emisor no son grandes en relación a la velocidad de la luz, este fenómeno puede ser descrito con la ecuación 2.1:

$$(2.1) \quad f = f_0 \cdot \left(\frac{c + v_r}{c + v_e} \right)$$

donde f es la frecuencia percibida por el emisor, f_0 la frecuencia original de la señal, v_r y v_e son las velocidades del receptor y emisor correspondientemente y c la velocidad de la luz. Es posible reescribir esta ecuación trasladando todo el movimiento al emisor:

$$(2.2) \quad f = f_0 \cdot \left(1 - \frac{v_{e,r}}{c} \right) = f_0 - f_0 \left(\frac{v_{e,r}}{c} \right)$$

donde la velocidad $v_{e,r} = v_e - v_r$ es la velocidad relativa entre el emisor y el receptor. Al valor $\delta f = -f_0 (v_{e,r}/c)$ se le conoce como corrimiento de Doppler, o “Doppler shift” en inglés, y corresponde a la frecuencia que varía la señal original producto del movimiento.

Relación señal a ruido

La relación señal a ruido, o **SNR** del inglés “signal-to-noise ratio“, corresponde a la relación entre la potencia de una señal con respecto a la del ruido. Se define como

$$(2.3) \quad SNR = \frac{P_{signal}}{P_{noise}}$$

donde P_{signal} es la potencia de la señal y P_{noise} la del ruido. Es habitual medir el **SNR** en decibeles, con lo cual, el **SNR** es:

$$(2.4) \quad SNR_{dB} = 10 \log_{10} \left(\frac{P_{signal}}{P_{noise}} \right) = P_{signal,dB} - P_{noise,dB}$$

para señales digitales existe un concepto muy útil que se desprende de la relación señal a ruido, que es la energía por bit en relación a la densidad espectral en potencia del ruido, definida como E_b/N_o . El factor E_b representa la energía por bit de la señal medida en Watts por bit ($[w/bit]$) y N_o es la densidad espectral de potencia del ruido medida en Watts por Hertz ($[w/Hz]$). La relación entre el **SNR** y el E_b/N_o está dada por la ecuación 2.5.

$$(2.5) \quad SNR = E_b/N_o \cdot \frac{f_b}{W}$$

donde f_b es la tasa de velocidad de datos (“bitrate”) medida en bits por segundo $[bits/s]$ y W el ancho de banda del canal (definido como la diferencia entre las frecuencias de corte máxima y mínima) medido en Hertz [Hz]. Esta relación también puede ser escrita en decibeles.

$$(2.6) \quad SNR_{dB} = 10 \log_{10}(E_b/N_o) + 10 \log_{10} \left(\frac{f_b}{W} \right)$$

2.1.2. Modulación digital

Es la etapa en donde el mensaje codificado se transforma en una señal análoga que se envía a través del canal. Se realiza modulando digitalmente una señal portadora análoga, la señal resultante contiene el mensaje digital, pero tiene propiedades en frecuencia de la señal análoga portadora. Existen diversos tipos de modulaciones digitales divididas de acuerdo a la modulación que producen en la onda resultante. Se explicará la modulación FSK mas en detalle, ya que corresponde a la utilizada en el esquema usado más adelante.

Modulación FSK

Es un tipo de modulación en el cual la onda resultante cambia la frecuencia de acuerdo al mensaje que se desea transmitir. En el caso del esquema mostrado en la figura 2.4, se aprecia que

la frecuencia de la señal modulada es mayor cuando el mensaje es 1 y menor cuando es 0. Este tipo de modulación se llama “Binary Frequency Shift Keying” (BFSK).

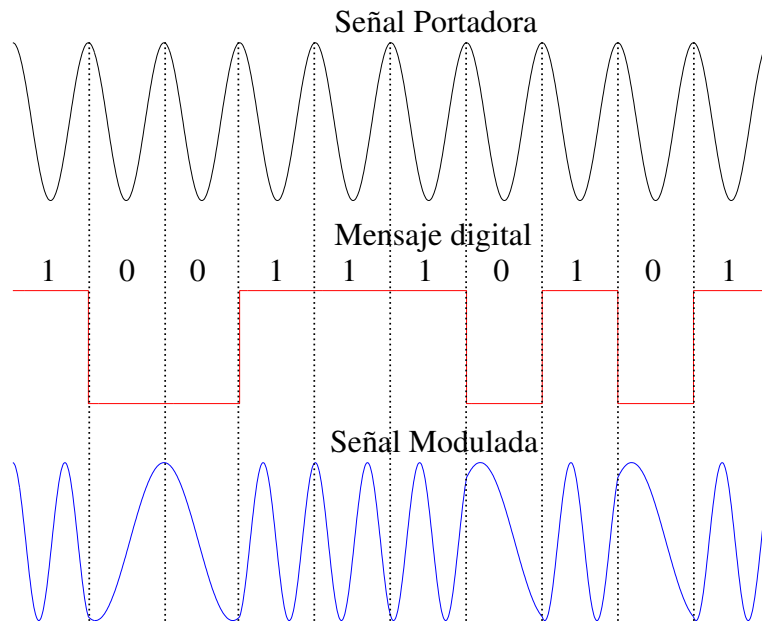


Figura 2.4: Esquema Modulación FSK

En parte, debido al proceso de modulación y demodulación digital se producen errores entre el mensaje enviado y el mensaje recibido. El desempeño teórico de una modulación digital es medido por la diferencia porcentual entre bits enviados correctamente y bits erróneos. Es decir:

$$(2.7) \quad BER = \frac{n_{err}}{n_{bits}}$$

donde n_{err} es el número de errores y n_{bits} el número de bits en la trama ocupada. Para el caso de la modulación FSK el desempeño teórico en relación al ruido del canal (E_b/N_0) se presenta en la figura 2.5.

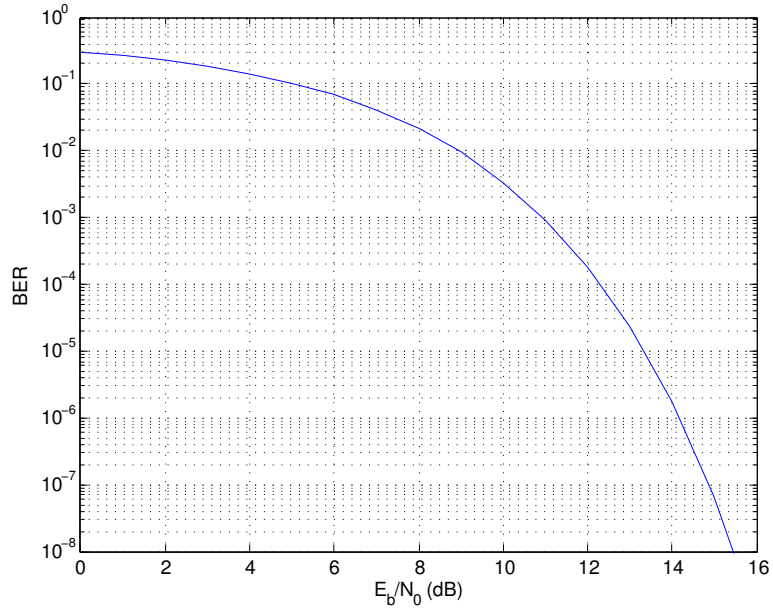


Figura 2.5: Curva de desempeño teórica de una modulación FSK no coherente utilizando un canal AWG

2.2. Procesamiento Digital de Señales

Concepto que se refiere a cualquier procesamiento de la señal que se realice a través de circuitos digitales. Esta técnica se ha hecho inmensamente popular en los últimos años debido a su bajo costo, en relación a los circuitos análogos, y a su capacidad de programación y adaptación. Para realizar el procesamiento digital de una señal es necesario tener un modelo de tiempo discreto que permita reconstruir la señal análoga.

2.2.1. Señales

Es posible representar una señal análoga como una función del tiempo y el espacio. Por ejemplo, segmentos de una señal de voz se pueden representar como una suma de múltiples sinusoides [24]. Es decir

$$(2.8) \quad s(t) = \sum_{i=1}^N A_i(t) \sin(2\pi F_i(t)t + \theta_i(t)).$$

donde $A_i(t)$, $F_i(t)$ y $\theta_i(t)$ son la amplitud, frecuencia y fase de cada senoide, correspondientemente. Así mismo es posible definir un “sistema” como un filtro que realiza una operación en la señal. Con esta definición, cualquier tipo de operación que afecte la señal, ya sea un

medio físico o un circuito digital, es considerado un sistema. El diagrama de bloques de un sistema general actuando sobre una señal análoga se muestra en la figura 2.6.

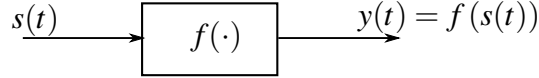


Figura 2.6: Esquema de un Sistema

Las señales se pueden clasificar, de acuerdo a su dependencia en el tiempo, en señales continuas o discretas. Una señal continua temporalmente tiene valores determinados para cada instante de un intervalo particular. Es decir

$$(2.9) \quad s(t) = f(t) \quad t \in [a, b]$$

donde el intervalo $[a, b]$ es el espacio temporal para el cual la señal se encuentra definida. Por otra parte, una señal discreta temporalmente solo tiene valores para instantes específicos en el tiempo. Si se usa el valor n para especificar un valor determinado en el tiempo, se tiene

$$(2.10) \quad s[n] = f[n] \quad n \in \mathbb{Z}.$$

De esta forma, una señal discreta puede ser representada como una secuencia de números reales o complejos. Si los tiempos t_n de una señal discreta son igualmente espaciados, es decir $t_n = nT_s$ para algún T_s , entonces es posible usar la notación $s(nT_s)$.

2.2.2. Teorema del muestreo

Es posible obtener una señal discreta en función de una señal continua. Para esto se obtienen los valores nT_s de la señal análoga y se produce una señal discreta tal que $s[n] = s_a(nT_s)$ para n en un rango tal que $nT_s \in [a, b]$. A este proceso se le llama “muestreo de una señal”. El valor T_s es el “periodo de muestreo” y el valor $1/T_s = F_s$ es la “frecuencia de muestreo”.

En alguna ocasiones es posible recuperar perfectamente la señal, a través de sus muestras, sin errores asociados al muestreo. Esto se realiza obteniendo la frecuencia máxima (F_{max}) que posee la señal análoga en el tiempo y escogiendo una frecuencia de muestreo tal que

$$(2.11) \quad F_s > 2F_{max} = 2B$$

donde B es el ancho de banda de la señal en banda base. Cuando se cumplen estas condiciones, es posible recuperar la señal ocupando la función de interpolación $g(x) = \sin(2\pi Bx)/2\pi Bx$.

$$(2.12) \quad s_a(t) = \sum_{n=-\infty}^{\infty} s[n] \cdot g\left(t - \frac{n}{F_s}\right).$$

donde $s[n] = s_a(n/F_s)$ es la señal digital y $s_a(t)$ es la señal recuperada por medio de las muestras. En la práctica, la suma hasta el infinito de la ecuación anterior es imposible de realizar, por lo que se escoge dos cotas, M_1 y M_2 , en las cuales se realiza esta suma:

$$(2.13) \quad s_a(t) = \sum_{n=-M_1}^{M_2} s[n] \cdot g\left(t - \frac{n}{F_s}\right).$$

2.2.3. Conversión de señal analógica a digital

La mayoría de las señales son análogas por naturaleza por lo que es necesario construir un conversor que haga la traducción entre el mundo análogo y el digital. En el esquema de la figura 2.7 se muestran las etapas de un conversor análogo-digital (A/D).

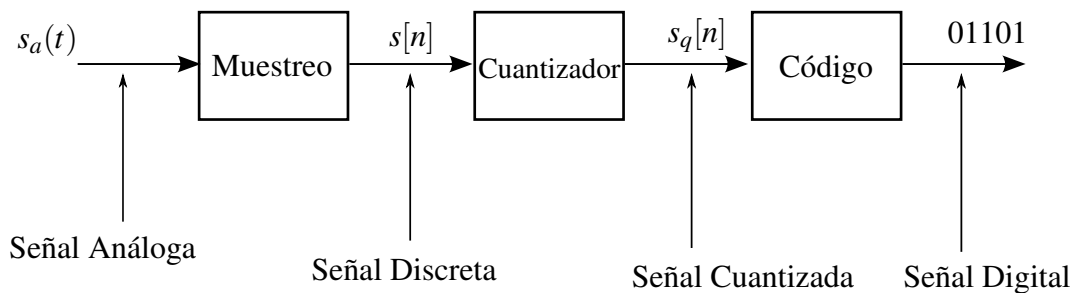


Figura 2.7: Diagrama de bloques conversor análogo-digital

- **Muestreo:** Se realiza la conversión de una señal análoga a una discreta en el tiempo. Esta etapa se realiza a través de un muestreador que obtiene el valor de la señal digital cada cierto período.
- **Cuantización:** Se convierte el valor continuo de la señal discretizada en el tiempo a un valor discreto, obteniéndose una señal de valor y tiempo discreto. A cada valor continuo de la señal $s[n]$ se le asigna un valor discreto $s_q[n]$ definido.
- **Codificación:** En esta etapa se obtiene una representación binaria para los valores discretos $s_q[n]$.

2.3. Comunicaciones Inalámbricas

Las comunicaciones inalámbricas son aquellas en las cuales el receptor y el emisor no se encuentran unidos por un medio de transmisión guiado, sino que se utilizan ondas electromagnéticas en el espacio libre para la comunicación entre las dos partes. Para comprender el funcionamiento de una comunicación inalámbrica es necesario entender la forma en que interactúan las ondas electromagnéticas entre el receptor, el medio y el emisor.

2.3.1. Ondas electromagnéticas

Si se tiene una antena transmisora fija en el espacio, es posible calcular el campo eléctrico producido por la antena para cualquier punto lejano. En el caso particular en que la señal transmitida sea una senoide de la forma $\cos(2\pi ft)$, se puede expresar el campo eléctrico, en el espacio y tiempo, con la ecuación 2.14

$$(2.14) \quad E(f, t, (r, \theta, \psi)) = \frac{\alpha_s(\theta, \psi, f) \cos(2\pi f(t - r/c))}{r}$$

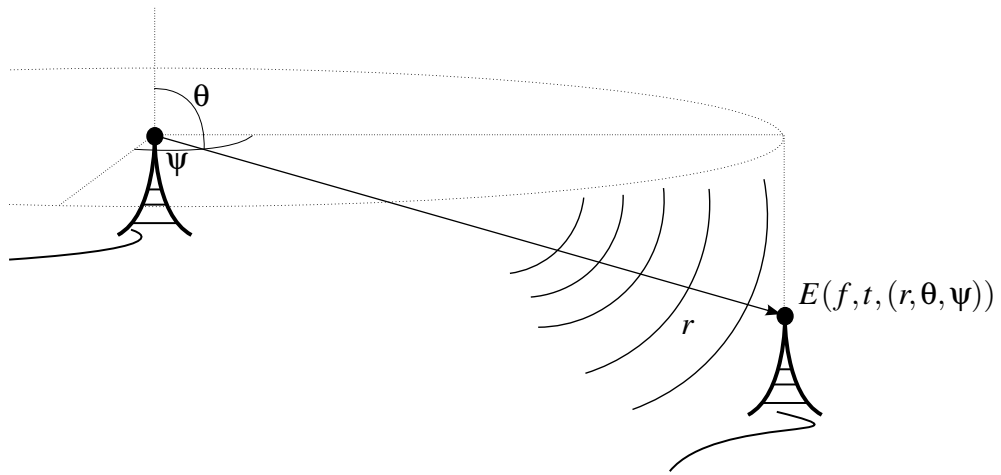


Figura 2.8: Propagación electromagnética entre dos puntos fijos.

donde el trío (r, θ, ψ) representa cualquier punto u en el espacio descrito por sus coordenadas esféricas. La constante c representa la velocidad de la luz, aproximadamente $3 \cdot 10^8 [m/s]$, y $\alpha_s(\theta, \psi, f)$ es la función que describe el patrón de radiación de la antena a una frecuencia f y en una dirección (θ, ψ) . Si se tiene una antena receptora en un punto u , el campo eléctrico que recibirá la antena será:

$$(2.15) \quad E(f, t, u) = \frac{\alpha_s(\theta, \psi, f) \cos(2\pi f(t - r/c))}{r}$$

En la figura 2.8 se muestra este esquema simple de comunicación inalámbrica en donde tanto el emisor como el receptor se encuentran fijos en el tiempo. Para esta configuración se tiene que las ecuaciones 2.14 y 2.15 son lineales en la entrada, lo que quiere decir que el campo eléctrico es una suma ponderada de la señal transmitida (una senoide). A este tipo de sistemas se les llama LTI (Lineal Time Invariant).

Si se cambia la configuración a una un poco mas compleja, en donde la antena receptora se encuentra en movimiento a una velocidad v , de forma que su posición en el espacio cambia en el

tiempo (como se muestra en la figura 2.9), se obtiene que el campo eléctrico en la antena receptora es.

$$(2.16) \quad E(f, t, (r_0 + vt, \theta, \psi)) = \frac{\alpha(\theta, \psi, f) \cos(2\pi f[(1 - v/c)t - r_0/c]}{r_0 + vt}$$

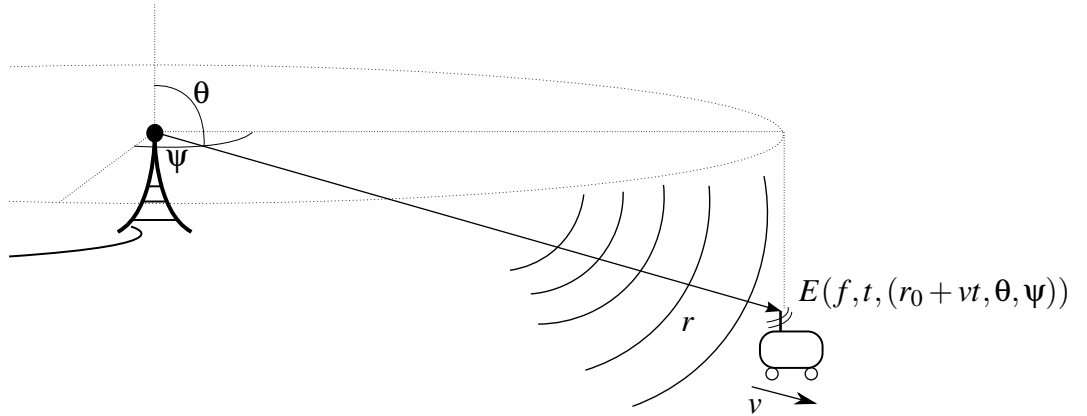


Figura 2.9: Propagación electromagnética entre emisor fijo y receptor en movimiento.

donde r_0 y v son la posición inicial y velocidad de la antena receptora. Se aprecia que la diferencia entre los campos 2.15 y 2.16 radica en el cambio de frecuencia producto del efecto Doppler (sección 2.1.1) que experimenta el campo eléctrico. Este cambio de frecuencia está dado por la ecuación 2.2 y se traduce en que el campo eléctrico ya no puede ser representado por un sistema LTI, debido a la atenuación cambiante en el denominador de la ecuación 2.16. Cuando la distancia, r , entre el emisor y el receptor es grande en comparación al factor vt , es posible describir el canal en términos de la entrada trasladada en frecuencia por un factor $-fv/c$.

El análisis anterior es independiente de si el movimiento proviene del receptor o emisor, y en el caso de que ambos se estén moviendo se puede realizar asumiendo un sistema de referencia estático con respecto al emisor.

2.3.2. Ondas electromagnéticas reflejadas

Debido a los fenómenos vistos en la sección 2.1.1 el campo eléctrico observado por la antena receptora al final del recorrido es la suma de varios campos desfasados, que son producidos principalmente por los fenómenos de reflexión, refracción y scattering. Para estos casos es posible generalizar la ecuación 2.16 de la siguiente forma

$$(2.17) \quad E(f, t, (r(t), \theta(t), \psi(t))) = \sum_i \alpha_i(r(t), \theta(t), \psi(t), f) \cos(2\pi f[t - \tau_i(f, t)])$$

donde los coeficientes $\alpha_i(r(t), \theta(t), \psi(t), f)$ son atenuadores específicos para cada campo eléctrico i y $\tau_i(f, t)$ es el desfase temporal en cada uno de ellos.

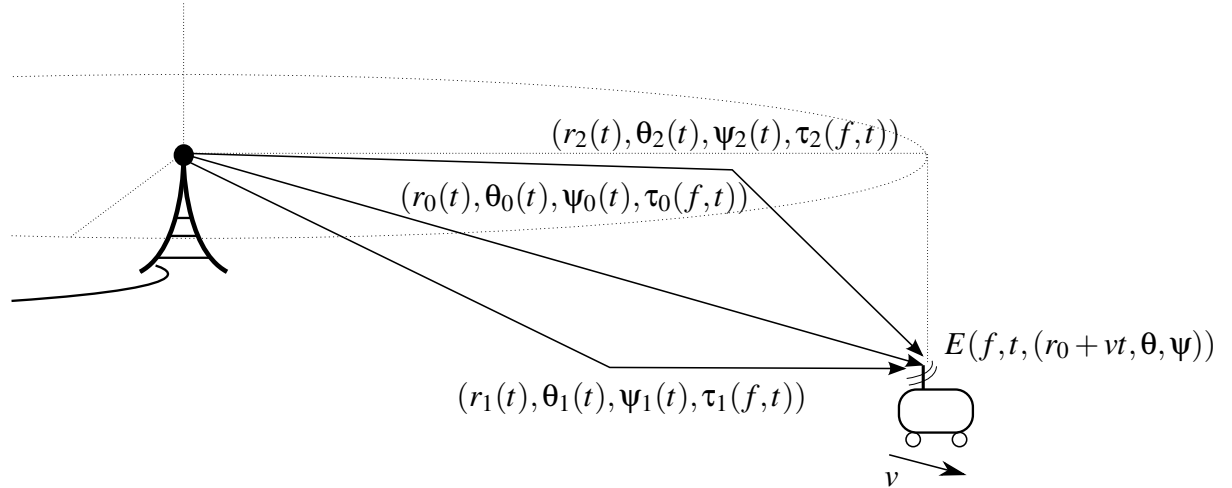


Figura 2.10: Propagación electromagnética con receptor móvil y múltiples caminos

2.3.3. Modelo para comunicaciones inalámbricas

En los casos anteriores se ha visto la forma del campo eléctrico para una entrada sinusoidal de la forma $\cos(2\pi f)$, llegando a la ecuación general del campo eléctrico (ecuación 2.17) para esta entrada en particular. No obstante, es posible generalizar este canal para cualquier par entrada-salida. De esta forma se obtiene

$$(2.18) \quad y(t) = \sum_i a_i(t, f) x(t - \tau_i(t, f))$$

donde $a_i(t, f)$ y $\tau_i(t, f)$ son los coeficientes de atenuación y retardo variantes en el tiempo, de cada camino i . Asumiendo que los coeficientes a_i y los retardos τ_i no dependen de la frecuencia f , el canal queda descrito de la siguiente forma:

$$(2.19) \quad y(t) = \sum_i a_i(t) x(t - \tau_i(t)).$$

Se puede representar este canal por su equivalente en banda base [28] a través de la siguiente ecuación:

$$(2.20) \quad y_b(t) = \sum_i a_i^b(t) x_b(t - \tau_i(t))$$

donde a_i^b , $y_b(t)$ y $x_b(t)$ son los equivalentes en banda base descritos por las siguientes ecuaciones

$$(2.21) \quad a_i^b(t) := a_i(t)e^{-j2\pi f_c \tau_i(t)}$$

$$(2.22) \quad y(t) = \sqrt{2}\Re[y_b(t)e^{j2\pi f_c t}]$$

$$(2.23) \quad x(t) = \sqrt{2}\Re[x_b(t)e^{j2\pi f_c t}]$$

2.3.4. Modelo discreto para comunicaciones inalámbricas

Para realizar simulaciones computacionales, es necesario formular un modelo discreto para el canal inalámbrico. Para esto se utiliza el teorema del muestreo. Asumiendo que el ancho de banda de la señal de entrada $x(t)$ está dado por un valor W , se tiene que la señal en banda-base, $x_b(t)$, tiene un ancho de banda $W/2 = B$ por lo que puede ser representada de la siguiente forma

$$(2.24) \quad x_b(t) = \sum_n x[n]g\left(t - \frac{n}{F_s}\right)$$

$$(2.25) \quad x_b(t) = \sum_n x[n]\text{sinc}(Wt - n)$$

donde $x[n] = x(n/W)$. Reemplazando la ecuación 2.25 en 2.20 se obtiene

$$(2.26) \quad y_b(t) = \sum_i a_i^b(t) \sum_n x[n]\text{sinc}(Wt - W\tau_i(t) - n)$$

luego, tomando la salida discreta $y[m]$ tal que $y[m] := y_b(m/W)$ y reemplazando en 2.26, se obtiene el modelo discreto definido por la ecuación 2.27.

$$(2.27) \quad y[m] = \sum_i a_i^b(m/W) \sum_n x[n]\text{sinc}[m - n - \tau_i(m/W)W]$$

haciendo el cambio de variable $\ell := m - n$ se obtiene

$$(2.28) \quad y[m] = \sum_i a_i^b(m/W) \sum_\ell x[m - \ell]\text{sinc}[\ell - \tau_i(m/W)W]$$

que también puede ser escrita en la forma

$$(2.29) \quad y[m] = \sum_\ell x[m - \ell] \sum_i a_i^b(m/W)\text{sinc}[\ell - \tau_i(m/W)W].$$

Finalmente, definiendo $h_\ell[m] := \sum_i a_i^b(m/W)\text{sinc}[\ell - \tau_i(m/W)W]$ se puede escribir el modelo discreto de forma más simple mediante la ecuación 2.30

$$(2.30) \quad y[m] = \sum_\ell h_\ell[m]x[m - \ell].$$

En el caso especial en que a_i y τ_i son invariantes en el tiempo se cumple que

$$(2.31) \quad h_\ell = \sum_i a_i^b \text{sinc}[\ell - \tau_i W].$$

El modelo de la ecuación 2.29 es conocido como filtro FIR (“finite impulse response”) o también como modelo TDL (“tap delay line”).

2.3.5. Ruido blanco aditivo Gaussiano

Es usual asumir la existencia de ruido blanco aleatorio Gaussiano, o AWGN por sus siglas en inglés “additive white gaussian noise”. Por esta razón, se agrega un término $\omega(t)$ al modelo de la ecuación 2.19 quedando de la siguiente forma

$$(2.32) \quad y(t) = \sum_i a_i(t)x(t - \tau_i(t)) + \omega(t)$$

donde $\omega(t)$ es una variable aleatoria Gaussiana independiente de densidad de potencia espectral $N_0/2$ (i.e, $E(\omega(0)\omega(t)) = N_0/2\delta(t)$) y media 0. Con este nuevo término el modelo discreto en banda base de la ecuación 2.30 queda descrito por la ecuación 2.33

$$(2.33) \quad y[m] = \sum_\ell h_\ell[m]x[m - \ell] + \omega[m]$$

donde $\omega[m]$ es el ruido después de haber pasado por un filtro pasa bajos en el instante $t = m/W$. La sucesión en el tiempo de $\omega[m]$ representa un proceso estocástico “complejo Gaussiano blanco de simetría circular”, lo cual quiere decir que las partes real e imaginaria del proceso son Gaussianas, de varianza $N_0/2$ y media 0, e independientes entre sí, además el proceso es independiente en el tiempo.

$$(2.34) \quad \omega[m] \sim \mathcal{CN}(0, N_0/2).$$

$$(2.35) \quad \begin{aligned} &\omega[m] = \omega_R[m] + \omega_I[m] \cdot j \\ \Rightarrow &\omega_R[m] \sim \mathcal{N}(0, N_0/2) \quad \omega_I[m] \sim \mathcal{N}(0, N_0/2).j. \end{aligned}$$

2.3.6. Tiempo de Coherencia

Tomando como referencia el modelo de la ecuación 2.30 el tiempo de coherencia se define (en sentido del orden de magnitud) como el tiempo que transcurre para que los coeficientes h_l cambien

significativamente en relación a su media. Más específicamente, se puede encontrar la siguiente relación [28]

$$(2.36) \quad T_c = \frac{1}{4D_s}$$

donde D_s es la dispersión de Doppler definida como $D_s := \max\left(f_c|\tau'_i(t) - \tau'_j(t)|\right)$ y $\tau'_i(t) = D_i/f_c$ con D_i es el corrimiento de Doppler en cada camino i .

En general, en los sistemas de comunicaciones inalámbricos es usual aproximar la dispersión de Doppler D_s como el corrimiento de Doppler máximo [11], es decir

$$(2.37) \quad T_c = \frac{1}{4D_s} = \frac{1}{4\max(D_i)} = \frac{1}{4f_c \frac{v_{max}}{c}}.$$

Sin embargo, esta aproximación no es adecuada para comunicaciones satelitales donde la velocidad radial es muy pequeña comparada a la tangencial [6]. Para estos casos es posible aproximar la dispersión de Doppler a través de la velocidad angular Ω [rad/s].

$$(2.38) \quad D_s = \sqrt{\left(\frac{\Omega}{d}\right)^2 + \left(\frac{V_g}{\Delta_c}\right)^2 + \left(\frac{1}{T_{ch}}\right)^2}$$

donde d es una constante dependiente de la estructura de los obstáculos en el canal, V_g es la velocidad del terminal terrestre, Δ_c una constante del orden de la longitud de onda de la señal y T_{ch} el tiempo de variación de los alrededores del móvil. Considerando una estación terrena estática la ecuación 2.38 se reduce a

$$(2.39) \quad D_s = \sqrt{\left(\frac{\Omega}{d}\right)^2}.$$

Si además no se considera obstáculo alguno en el canal, la dispersión de Doppler dependerá directamente de la velocidad angular del satélite, que es en general baja, menor a 10^{-2} radianes por segundo. Por esta razón la dispersión de Doppler para comunicaciones satelitales (fundamentalmente en órbitas de tipo GEO, LEO o MEO), dependerá casi exclusivamente del movimiento de la antena terrestre [4] [12] [27] y en el caso de que no exista será un valor bajo que en la práctica no supera los 3 [Hz] [6].

2.3.7. Modelos estadísticos de canales

Estos modelos caracterizan estadísticamente la forma en que varían los coeficientes $h_l[m]$ en el modelo discreto (ecuación 2.30). Esto significa conocer cuantos “taps” son necesarios, que tan rápido varían en el tiempo y la magnitud de este cambio.

En la práctica, los modelos estadísticos, se utilizan como una forma de simular las alteraciones que se producen en un canal real. Existen gran cantidad de estos modelos, pero para comunicaciones inalámbricas se suele utilizar los modelos de Rayleigh o Rician.

Desvanecimiento de Rayleigh

Se basa la suposición de que existen un gran número de caminos cada uno con ganancia a_i aleatoria. Por esta razón es razonable suponer que la fase de cada camino $\phi_i = 2\pi\tau_i$ sea una variable aleatoria uniformemente distribuida entre 0 y 2π [28].

$$(2.40) \quad 2\pi\tau_i = \phi_i \sim UNIFORME(0, 2\pi)$$

$$(2.41) \quad f(\phi_i) = \begin{cases} \frac{1}{2\pi} & \text{para } 0 \leq \phi_i \leq 2\pi, \\ 0 & \text{para } \phi_i < 0 \text{ o } \phi_i > 2\pi, \end{cases}$$

El desvanecimiento de Rayleigh también asume que la contribución de cada camino i ($a_i^b(m/W)\text{sinc}[l - \tau_i(m/W)W]$) puede ser modelada como un proceso aleatorio Gaussiano blanco de simetría circular, por lo tanto, cada “tap” $h_\ell[m]$, es una suma de muchas variables aleatorias circulares simétricas independientes, de esta forma gracias al teorema central del límite, se llega a la conclusión que $h_\ell[m]$ es también un proceso aleatorio de simetría circular, con media 0 y varianza σ_ℓ^2 .

$$(2.42) \quad h_\ell[m] \sim \mathcal{CN}(0, \sigma_\ell^2),$$

y, por lo tanto, la magnitud $|h_\ell[m]|$, de cada “tap” es una variable aleatoria con distribución de Rayleigh.

$$(2.43) \quad |h_\ell[m]| \sim \mathcal{Ray}(\sigma_\ell^2)$$

$$(2.44) \quad f(|h_\ell[m]|; \sigma_\ell) = \frac{|h_\ell[m]|}{\sigma_\ell^2} e^{-|h_\ell[m]|^2/2\sigma_\ell^2}$$

Desvanecimiento Riciano

Otro modelo, alternativo al modelo de Rayleigh, se desprende asumiendo que existe un camino con línea de visión (LOS) que es comparativamente mas grande con respecto a la magnitud de potencia de los otros caminos. En este caso se tiene que, por lo menos para un valor de ℓ , $h_\ell[m]$ es

$$(2.45) \quad h_\ell[m] = \sqrt{\frac{K}{K+1}} \sigma_\ell e^{j\theta} + \sqrt{\frac{1}{K+1}} \mathcal{CN}(0, \sigma_\ell^2)$$

donde el primer termino corresponde al camino LOS con fase uniforme θ y el segundo termino es una suma de los caminos restantes. La constante K es conocida como el factor-k y representa la

proporción de energía entre el camino LOS y los restantes. En este caso, se tiene, que la magnitud de la variable $h_\ell[m]$ sigue una función de distribución Riciana.

$$(2.46) \quad |h_\ell[m]| \sim \mathcal{R}_{ice}(v, \sigma)$$

$$(2.47) \quad f(|h_\ell[m]|; v, \sigma) = \frac{|h_\ell[m]|}{\sigma^2} \exp\left(-\frac{(|h_\ell[m]|^2 + v^2)}{2\sigma^2}\right) I_0\left(\frac{|h_\ell[m]|v}{\sigma^2}\right)$$

en donde I_0 es la función de Bessel modificada de orden 0 y los estadísticos v y σ tienen relación con las funciones de distribución de la parte real e imaginaria de $h_\ell[n]$:

$$(2.48) \quad h_\ell[m] = h_{real}[m; \ell] + h_{img}[m; \ell]j$$

$$h_{real}[m; \ell] \sim \mathcal{N}(v \cos(\theta), \sigma^2); \quad h_{img}[m; \ell] \sim \mathcal{N}(v \sin(\theta), \sigma^2) \quad \forall \theta.$$

Es posible realizar un cambio de variable de v y σ a K y Ω , en donde la variable K es el factor-k nombrado anteriormente y la variable Ω representa la potencia de $|h_\ell|$. La función de distribución en función de las nuevas variables esta definida en la ecuación 2.49:

$$(2.49) \quad f(|h_\ell[m]|; K, \Omega) = \frac{2(K+1)|h_\ell[m]|}{\Omega} \exp\left(-K - \frac{(K+1)|h_\ell[m]|^2}{\Omega}\right) I_0\left(2\sqrt{\frac{K(K+1)}{\Omega}}|h_\ell[m]|\right)$$

en donde $v^2 = \frac{K}{1+K}\Omega$ y $\sigma^2 = \frac{\Omega}{2(1+K)}$.

Estacionario en el sentido amplio

Un proceso se denomina estacionario cuando su función de distribución conjunta no cambia ni en el tiempo ni en el espacio. Consecuentemente, sus estadísticos tampoco varían. El proceso se considera estacionario en el sentido amplio, o “wide sense stationary” (WSS), cuando sus primeros dos momentos no cambian en el tiempo.

$$(2.50) \quad E\{x(t)\} = m_x(t) = m_x(t + \tau) \quad \text{para todo } \tau \in \mathbb{R}$$

$$(2.51) \quad E\{x(t_1)x(t_2)\} = R_x(t_1, t_2) = R_x(t_1 + \tau, t_2 + \tau) = R_x(t_1 - t_2, 0) \quad \text{para todo } \tau \in \mathbb{R}$$

donde $m_x(t)$ es la media de $x(t)$ y $R_x(\tau)$ la covarianza.

2.4. Comunicaciones satélites

Un satélite de comunicaciones es un módulo artificial en órbita, que es usado para transmitir y recibir señales desde de la tierra. Los satélites se complementan con una estación terrena, que

son usadas para transmitir y recibir señales hacia el espacio. En un esquema satelital pueden existir muchas estaciones terrenas comunicándose con un mismo satélite o una estación comunicándose con varios satélites. El esquema de una comunicación satelital típica se muestra en la figura 2.11.

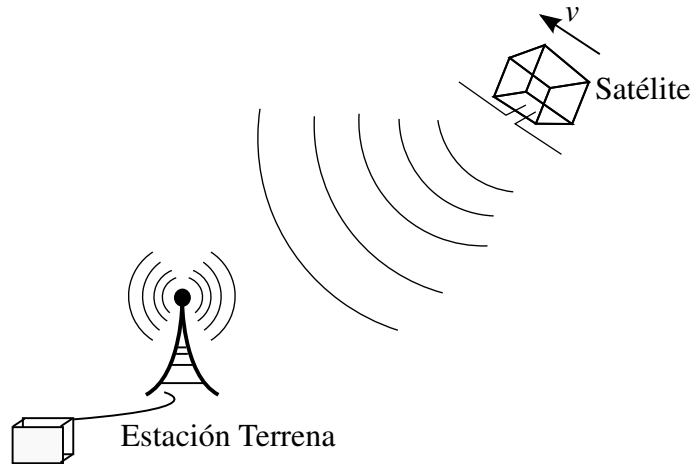


Figura 2.11: Esquema de comunicación por satélite

2.4.1. Tipos de órbita

Es posible dividir los tipos de canales por la forma de órbita con respecto a la tierra. El tipo de órbita es dado por la distancia entre el satélite y el centro de masa del planeta.

Órbita Geoestacionaria

Una órbita geoestacionaria, o **GEO** por sus siglas en inglés, es elegida de tal forma que el periodo de la órbita coincida con el periodo de la tierra. De esta forma, un observador en la tierra verá el satélite como un punto fijo en el espacio. Para obtener una órbita de este tipo es necesario fijar un radio específico entre el satélite y la tierra. Este radio se puede calcular a través de la tercera ley de Kepler

$$(2.52) \quad r_s = \left[\frac{GM_T}{4\pi^2} \right]^{\frac{1}{3}} T^{\frac{2}{3}}$$

donde G es la constante de gravitación universal, M_T la masa de la tierra y T la media del periodo de la tierra, que corresponde a 86.164,09 segundos. Reemplazando por los valores adecuados en cada caso, se obtiene un radio de alrededor de 42.000 kilómetros. Si a éste valor se le resta el radio de la tierra (6.370[km]) se obtiene una distancia h_s aproximada de 36.000[km] que es el valor que se usa habitualmente para cálculos orbitales geoestacionarios. En la práctica, la órbita geoestacionaria perfecta es irrealizable y se utiliza un sistema de control, ya sea desde la estación

terrena o desde el satélite, que corrige los errores de sincronización que existan en la órbita. La mayoría de los satélites de comunicación operan en éste radio orbital, ya que presenta muchas ventajas en el aspecto de cobertura y seguimiento. Sin embargo, debido a estas ventajas, existe una gran demanda por el uso de estos lugares geoestacionarios que se encuentran limitados al radio anteriormente calculado. Con el fin de optimizar este espacio limitado, existe una organización (La "International Telecommunication Union" o ITU) que regula el uso de estos espacios. Las características generales de un satélite de tipo GEO son presentadas la tabla 2.1.

Tipo de satélite	GEO
Tiempo de exposición	Total
Seguimiento	Ninguno o muy poco
Cobertura	2 o 3 satélites para cobertura global
Usos	Comunicación, Broadcasting

Tabla 2.1: Características de satélites Geoestacionarios

Órbita de baja distancia

Los satélites que orbitan, mucho mas cercanos a la tierra que los GEO, comúnmente a una altitud entre los 160 y los 2500 kilómetros desde la superficie de la tierra son llamados satélites de baja órbita o satélites de tipo LEO por sus siglas en inglés. Estos satélites viajan a una velocidad mucho mayor que los GEO y presentan muchas desventajas frente a estos, siendo la mayor el tiempo de exposición en relación a un punto determinado en la tierra, que es de alrededor de 8 minutos. Sin embargo los satélites tipo LEO se han considerado para proyectos de redes móviles, donde la potencia necesaria para establecer la comunicación es menor y se puede tener una configuración de muchos satélites pequeños y más baratos que otorguen cobertura mundial. Algunas características de estos satélites se presentan en la tabla 2.2.

Tipo de satélite	LEO
Tiempo de exposición	8 minutos por pasada
Seguimiento	requiere de un terminal que siga al satélite
Cobertura	esquemas de (12, 24, 66) satélites para cobertura global
Usos	investigación, redes móviles

Tabla 2.2: Características satélites de baja órbita

Órbita de mediana distancia

Los satélites que operan en rangos de altitudes entre los satélites de tipo LEO y GEO son llamados satélites de mediana órbita o satélites de tipo MEO por sus siglas en inglés. Usualmente

operan a una altitud entre los 10.000 y 20.000 kilómetros. Este tipo de satélite ha probado ser útil para fines de navegación o determinación de posiciones. La red **GPS** está formada por satélites de tipo **MEO**, y ocupa una constelación de 24 satélites a una altitud de 20.184km. Algunas características de esta órbita son presentadas en la tabla 2.3

Tipo de satélite	MEO
Tiempo de exposición	1 a 2 horas por pasada
Seguimiento	Similar al LEO pero menos brusco
Cobertura	GPS usa 24 satélites para cobertura global
Usos	meteorológicos, posicionamiento global

Tabla 2.3: Características satélites de mediana órbita

Órbita altamente elíptica

Los satélites que operan en órbitas altamente elípticas, o satélites de tipo **HEO** por sus siglas en inglés, son usados principalmente para establecer cobertura a latitudes altas o bajas que no estén cubiertas por un satélite de tipo **GEO** y en aquellas donde se requiera un mayor tiempo de exposición que el ofrecido por satélites de tipo **LEO**. A diferencia de las otras órbitas vistas, en este caso, no se tiene una órbita casi circular, si no que una altamente excéntrica, por lo que es posible diferenciar el apogeo del perigeo. El perigeo es la distancia mas pequeña que alcanza el satélite con respecto a la tierra y en la que alcanza mayor velocidad. El apogeo es la distancia más grande de la órbita y es la posición que menor velocidad alcanza. Algunas características de estas órbitas se presentan en la tabla 2.4

Tipo de satélite	HEO
Tiempo de exposición	Dependiente de la posición del Satélite
Seguimiento	En el radio mayor necesita menos que el LEO o MEO
Cobertura	Latitudes altas y bajas
Usos	Alcance de Latitudes altas (Molniya)

Tabla 2.4: Características satélites de órbita altamente elípticas

2.4.2. Satélites miniaturizados

El término satélite miniaturizado se refiere a un satélite artificial de proporciones pequeñas y peso reducido (Usualmente bajo los 500 [kg]), lo que se traduce a un menor precio de construcción y lanzamiento del mismo. La variable más significativa en el precio del satélite es el peso, debido principalmente a lo elevado que es el costo del lanzamiento. Existen múltiples clasificaciones para la diferenciación de estos satélites en comparación con los normales, algunas se basan en el objetivo

del satélite y otras en el peso. Un standar utilizado para el diseño construcción de estos satélites miniaturizados es el "standar cubesat"

2.4.3. Satélites tipo "cubesat"

Es un tipo de satélite miniaturizado estandarizado que tiene un volumen de 1 litro (cubo de 10 cm por lado) y masa no mayor a 1.33 kilogramos. Este estandar se desarrolló a principios de 1999 en las Universidades Politécnica de California y de Standford, donde fue creado para ayudar a las universidades que quisieran desarrollar ciencia en el espacio. Normalmente los "cubesat" son construidos para orbitar a bajas distancias (órbitas tipo LEO o MEO) y grandes velocidades, lo que hace que la ventana de tiempo en que estos satélites son visibles desde un punto en la tierra sea pequeño (del orden de minutos) por lo que es de suma importancia la transmisión de datos con el menor error y mayor tasa posible.

2.4.4. Proyecto SUCHAI

Las siglas **SUCHAI**, vienen del inglés y significan "Satellite of University of Chile for Aerospace Investigation". Se trata de un proyecto, de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile, para poner en órbita el primer satélite chileno armado y diseñado por estudiantes universitarios.

Los objetivos del proyecto son tanto académicos como educacionales y técnicos. Por una parte el objetivo educacional es evaluar los beneficios académicos y curriculares de un proyecto complejo y de larga duración como el propuesto. El objetivo técnico es integrar y desarrollar los componentes del satélite junto con sus sistemas de operación en la Universidad de Chile, ganando la pericia y los conocimientos logísticos requeridos y por último, los objetivos científicos corresponden a posibles experimentos que puedan implementarse una vez que el satélite se encuentre en órbita. Para realizar este proyecto de gran envergadura se ha dividido el trabajo en diversas áreas cada una enfocada a diferentes subsistemas del satélite. Estas áreas, aunque diferentes entre sí, son totalmente complementarias y vitales para el proyecto en su totalidad. Las áreas son: **Comunicaciones, Energía, Procesamiento de Datos, Mecánica y Payloads**. En particular, el área en donde tiene aplicación el tema de memoria, es el área de Comunicaciones, que es la encargada de lograr la comunicación, entre el satélite y la estación terrena, y se encarga de la construcción y posicionamiento de esta.

Para el primer satélite a lanzar se usará el estandar "cubesat" de 1 unidad visto en la sección [2.4.3](#)

Esquema de comunicación del proyecto

Para el caso particular del proyecto, se utilizará el esquema específico mostrado en la figura 2.12.

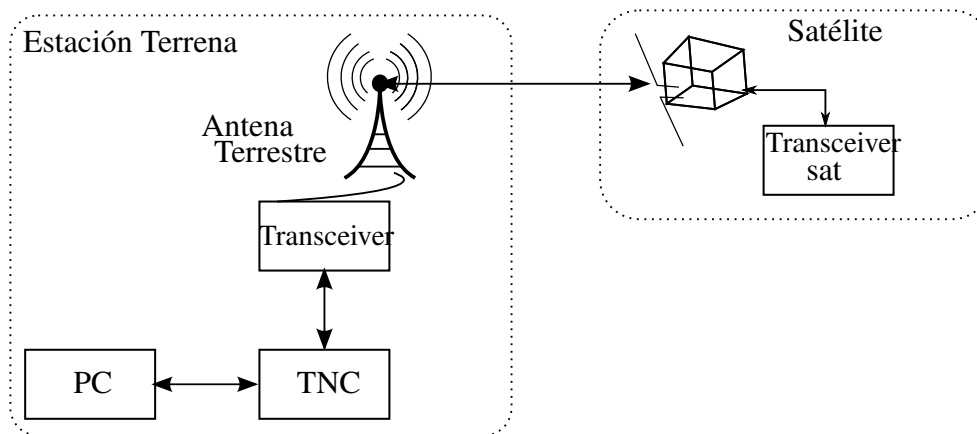


Figura 2.12: Esquema de comunicaciones del proyecto [SUCHAI](#). En la estación terrena; el bloque PC corresponde a un servidor que realiza el almacenamiento y manejo de datos, TNC realiza la comunicación binaria entre el PC y el transceiver que transmite y recibe la señal análoga a través de la antena terrestre. En la parte satelital se recibe y envía la señal análoga a través de la antena y transceiver.

Los elementos que conforman el esquema de comunicaciones del proyecto se describen a continuación:

1. **Computador servidor (PC):** Realiza la función de administrar y almacenar datos binarios, así como también el seguimiento del satélite y movimiento de los rotores. Recibe o envía un mensaje binario a través del puerto serial, que se encuentra conectado al TNC. El computador se encuentra conectado a la red local y su puerto serial compartido con otros computadores, por lo que es posible ocupar cualquier computador en la red como emisor o receptor del mensaje.
2. **“Terminal node controller” (TNC):** Es el dispositivo que realiza la comunicación, a través de un protocolo particular, entre el computador servidor y el transceiver terrestre. El TNC del proyecto ocupa el protocolo KISS para su funcionamiento.
3. **Transceiver terrestre:** Es el dispositivo que realiza la conversión entre la señal digital y la análoga, y envía o recibe esta señal por medio de un cable coaxial conectado a la antena. El esquema usado en el proyecto ocupa un ICOM-910 con una modulación FSK de 1200 [bit/s] trabajando en frecuencias UHF entre los 430 a 440 [MHz].

4. **Antena terrestre:** Es un elemento pasivo que direcciona la señal enviada por el transceiver. El esquema del proyecto ocupa una antena Yagi cruzada con un boom de 5 metros y 42 elementos parásitos. Antes de la antena es posible conectar un amplificador de bajo ruido para aumentar la potencia de las señales recibidas.
5. **Medio de transmisión:** Corresponde al espacio libre por el que viaja la señal entre el satélite y la estación terrena. Debido al seguimiento que se realiza desde tierra se espera que siempre existe una línea de vista para los períodos en que la comunicación es posible. Se esperan niveles de SNR de alrededor de los 10[dB] para el enlace de bajada y 15 [dB] para el enlace de subida [1]
6. **Antena satelital:** Es la antena que recibe y envía la señal análoga dentro del satélite. Corresponderá a un dipolo adaptado a UHF conectado al transceiver satelital.
7. **Transceiver satelital:** Realiza las mismas funciones que el transceiver terrestre, pero dentro del satélite.

2.5. Estimación estadística de señales

La estimación estadística de señales se basa en determinar el valor de un grupo de parámetros que definen el comportamiento de una señal determinada. Formalizando este concepto, se tiene una variable aleatoria $x[n]$ que simboliza una señal en el tiempo y se asume que esta variable tiene una función de distribución que depende de alguna forma de los valores del vector θ a estimar. Es decir

$$(2.53) \quad PDF(x[n]) = p(x[n]; \theta)$$

donde θ es un vector de los parámetros a estimar y $x[n]$ la variable aleatoria en el tiempo n . Conociendo el valor de esta función de distribución, se desea obtener un estimador $\hat{\theta}$ cuyo valor sea similar a θ , tal que $\hat{\theta} = f(x[n])$. Es posible intuir que existen una cantidad infinita de posibles estimadores, por lo que la elección de uno en particular no es una decisión trivial. En la práctica, existen ciertas normas que aseguran la calidad del estimador, como el sesgo estadístico o su varianza.

2.5.1. Calidad del estimador

Para analizar la calidad de un estimador $\hat{\theta}$ se debe tener en cuenta que éste se trata de una variable aleatoria y que, por lo tanto, tiene sentido calcular su esperanza y varianza. En este sentido, es intuitivo pensar que para que el estimador sea de calidad, es necesario que la esperanza de éste sea θ y la varianza sea la menor posible. A partir de lo anterior se obtienen los conceptos de estimador insesgado y de mínima varianza.

Estimador Insesgado

Un estimador insesgado, o “unbiased” en inglés, es aquel cuya esperanza corresponde al valor a estimar. Es decir

$$(2.54) \quad E(\hat{\theta}) = \int_{-\infty}^{\infty} \hat{\theta} f(\hat{\theta}) d\hat{\theta} = \theta$$

se define el sesgo como $b(\theta)$ tal que

$$(2.55) \quad b(\theta) = E(\hat{\theta}) - \theta$$

y un estimador se considera insesgado cuando $b(\theta)$ es 0.

Criterio de mínima varianza

El criterio de mínima varianza se desglosa al minimizar el error cuadrático medio (**MSE**), que se define de la siguiente forma

$$(2.56) \quad mse(\hat{\theta}) = E [(\hat{\theta} - \theta)^2]$$

es posible reescribir el **MSE** tal que

$$\begin{aligned} mse(\hat{\theta}) &= E \left[[(\hat{\theta} - E(\hat{\theta})) + (E(\hat{\theta}) - \theta)]^2 \right] \\ mse(\hat{\theta}) &= var(\hat{\theta}) + [E(\hat{\theta}) - \theta]^2 \\ (2.57) \quad mse(\hat{\theta}) &= var(\hat{\theta}) + b^2(\theta) \end{aligned}$$

donde se aprecia que el error cuadrático medio depende de la varianza del estimador y el sesgo de éste. En el caso que el estimador sea insesgado, se tiene que $b(\theta) = 0$ y minimizar el **MSE** es equivalente a minimizar la varianza. Se dice que un estimador $\hat{\theta}$ es **MVU** (“Minimum Variance Unbiased”) si es insesgado y tiene la mínima varianza alcanzable.

2.5.2. Cota inferior de Cramer-Rao

La cota de Cramer-Rao fija teóricamente la varianza mínima que puede obtener un estimador dada una función de distribución en particular. Para algunos tipos de señales, no solo establece un límite inferior para la varianza, sino que también define el estimador que, teóricamente, llega a obtener ese valor. Si se asume que la señal x tiene una función de distribución $p(x; \theta)$ que satisface la condición de regularidad definida en la ecuación 2.58.

$$(2.58) \quad E \left[\frac{\partial^2 \ln(p(x; \theta))}{\partial \theta^2} \right] = 0 \quad \text{para todo } \theta$$

entonces la varianza de cualquier estimador insesgado debe satisfacer la inecuación 2.59

$$(2.59) \quad \text{var}(\hat{\theta}) \geq \frac{1}{-E \left[\frac{\partial^2 \ln(p(x; \theta))}{\partial \theta^2} \right]}$$

y es posible encontrar un estimador $\hat{\theta}$ insesgado si y solo si

$$(2.60) \quad \frac{\partial \ln(p(x; \theta))}{\partial \theta} = I(\theta)(g(x) - \theta)$$

para alguna función g e I . En el caso que se cumpla lo anterior, se tiene que $\hat{\theta} = g(x)$ y la mínima varianza es $1/I(\theta)$. Sin embargo, para muchos casos no es posible o es muy complicado encontrar un estimador que cumpla el límite de Cramer-Rao y es necesario encontrar una solución sub-óptima. La cota de la ecuación 2.59 se puede generalizar para un caso vectorial donde $\theta = [\theta_1 \theta_2 \dots \theta_p]^T$ y cada θ_i cumple la condición de regularidad mostrada en la ecuación 2.58. Para este caso se tiene que todo vector de estimadores $\hat{\theta}$ debe satisfacer

$$(2.61) \quad \mathbf{C}_{\hat{\theta}} - \mathbf{I}^{-1}(\theta) \geq 0$$

donde la expresión ≥ 0 quiere decir que la matriz es semi-definida positiva. La matriz $\mathbf{I}(\theta)$ es la matriz de Fisher, definida como

$$(2.62) \quad [\mathbf{I}^{-1}(\theta)]_{ij} = -E \left[\frac{\partial^2 \ln(p(x; \theta))}{\partial \theta_i \partial \theta_j} \right]$$

de igual forma que su símil escalar, es posible, en algunos casos, encontrar un estimador que logre la cota $\mathbf{C}_{\hat{\theta}} = \mathbf{I}^{-1}(\theta)$ si y solo si

$$(2.63) \quad \frac{\partial \ln(p(x; \theta))}{\partial \theta} = \mathbf{I}(\theta)(\mathbf{g}(x) - \theta)$$

para alguna función p -dimensional \mathbf{g} . Si se logra lo anterior, entonces el estimador **MVU** está dado por $\hat{\theta} = \mathbf{g}(x)$.

2.5.3. Modelos Lineales

En general, encontrar el estimador **MVU** no es sencillo [16]; Sin embargo, es posible generalizar una forma de obtener este estimador para una clase específica de modelos. Éstos son los llamados modelos lineales, en donde la obtención del **MVU** es posible y definida fácilmente. Se define como “modelo lineal” a un modelo en que el valor de la señal observada para cada instante de tiempo n , $x[n]$, se encuentra descrito por la siguiente ecuación vectorial:

$$(2.64) \quad \mathbf{X} = \mathbf{H}\theta + \mathbf{W}$$

donde $\mathbf{X} = [x[0] \dots x[N]]^T$ es un vector de $(Nx1)$ observaciones, \mathbf{H} es una matriz, de valores conocidos, de dimensiones $(Nx p)$ ($N > p$), θ es un vector de $(p \times 1)$ que contiene los parámetros a estimar y $\mathbf{W} = [\omega[0] \dots \omega[N]]^T$ es un vector de ruido de $(Nx1)$ con función de distribución $\mathcal{N}(0, \sigma^2 \mathbf{I})$. Para el modelo de la ecuación 2.64 se tiene que la función de distribución $p(x[n]; \theta)$ para cada instante n tiene la siguiente forma

$$(2.65) \quad p(x[n]; \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x[n] - (\mathbf{H}\theta)[n])^2}{2\sigma^2}}.$$

Se define la ecuación 2.68 de la siguiente forma

$$(2.66) \quad \frac{\partial \ln(p(\mathbf{X}; \theta))}{\partial \theta} = \frac{\partial}{\partial \theta} \left[-\ln(2\pi\sigma^2)^{N/2} - \frac{1}{2\sigma^2} (\mathbf{X} - \mathbf{H}\theta)^T (\mathbf{X} - \mathbf{H}\theta) \right]$$

derivando y utilizando algunas identidades algebraicas [16], se obtiene

$$(2.67) \quad \frac{\partial \ln(p(\mathbf{X}; \theta))}{\partial \theta} = \frac{1}{\sigma^2} [\mathbf{H}^T \mathbf{X} - \mathbf{H}^T \mathbf{H} \theta]$$

donde la función $p()$ es N -dimensional y denota la distribución de cada señal $x[n]$. Factorizando por $\frac{\mathbf{H}^T \mathbf{H}}{\sigma^2}$ se obtiene

$$(2.68) \quad \frac{\partial \ln(p(\mathbf{X}; \theta))}{\partial \theta} = \frac{\mathbf{H}^T \mathbf{H}}{\sigma^2} [(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{X} - \theta]$$

donde se aprecia una forma similar a la descrita en la ecuación 2.63. Identificando los valores $g(\mathbf{X})$ y $\mathbf{I}(\theta)$ se obtiene

$$(2.69) \quad \hat{\theta} = g(\mathbf{X}) = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x}$$

$$(2.70) \quad \mathbf{I}(\theta) = \frac{\mathbf{H}^T \mathbf{H}}{\sigma^2}$$

donde el estimador MVU del sistema lineal esta dado por la ecuación 2.69 y su covarianza es

$$(2.71) \quad \mathbf{C}_{\hat{\theta}} = \mathbf{I}^{-1}(\theta) = \sigma^2 (\mathbf{H}^T \mathbf{H})^{-1}.$$

Adicionalmente, el MVU encontrado en 2.69 alcanza la cota de Cramer-Rao(CRLB).

2.5.4. Estimador de máxima verosimilitud

Cuando el MVU no existe o es muy complejo de calcular es necesario encontrar un estimador sub-óptimo. Una opción popular para estos casos, por su facilidad y buenos resultados con muchas observaciones [16], es el estimador de máxima verosimilitud (MLE) que se basa en maximizar la

función de verosimilitud (“likelihood” en inglés) $p(\mathbf{x}; \theta)$. Para un vector de parámetros θ a estimar, asumiendo que la función de máxima verosimilitud es diferenciable, el **MLE** se encuentra con la siguiente ecuación

$$(2.72) \quad \left. \frac{\partial \ln(p(\mathbf{X}; \theta))}{\partial \theta} \right|_{\theta=\hat{\theta}} = 0;$$

si existen múltiples soluciones, entonces el valor que maximice la función será el **MLE**. En el caso de un sistema lineal, como el estudiado en la sección anterior, es directo, al igualar la ecuación 2.68 a cero, que el **MLE** es también el estimador **MVU** visto en la ecuación 2.69.

2.5.5. Cálculo secuencial del estimador lineal

El procesamiento de señales normalmente se realiza sobre una señal continua que está siendo muestreada en el tiempo a medida que se recibe la señal. La estimación de los valores requeridos puede hacerse esperando la llegada de todos los datos necesarios o secuencialmente a medida que se van recibiendo. Esta segunda posibilidad a menudo presenta ventajas en la implementación física del estimador y en la complejidad del algoritmo a realizar.

Asumiendo un modelo lineal y ruido blanco Gaussiano el vector de la señal observada hasta un tiempo n , $\mathbf{x}[n]$ queda definido por la ecuación

$$(2.73) \quad \mathbf{X}[n] = \mathbf{H}[n]\theta + \mathbf{W}[n]$$

$$(2.74) \quad \mathbf{X}[n] \rightarrow (n \times 1) \quad \mathbf{H}[n] \rightarrow (n \times p) \quad \mathbf{W}[n] \rightarrow (n \times 1)$$

donde $\mathbf{H}[n]$ es la matriz de observaciones que se tiene hasta el tiempo n , θ el vector de parámetros a estimar y $\mathbf{W}[n]$ un vector con ruido blanco Gaussiano. Es posible descomponer la matriz de observaciones y el vector de la señal observada de la siguiente forma:

$$(2.75) \quad \mathbf{H}[n] = \begin{bmatrix} \mathbf{H}[n-1] \\ \mathbf{h}^T \end{bmatrix}$$

$$\mathbf{X}[n] = [x[0] \quad x[1] \quad \dots \quad x[n]]^T$$

donde n es la última observación obtenida. Con esto el estimador lineal de la ecuación 2.69 puede ser reescrito de la siguiente forma

$$(2.76) \quad \begin{aligned} \hat{\theta}[n] &= (\mathbf{H}^T[n]\mathbf{H}[n])^{-1}\mathbf{H}^T[n]\mathbf{X}[n] \\ &= \left(\begin{bmatrix} \mathbf{H}^T[n-1] & \mathbf{h} \end{bmatrix} \begin{bmatrix} \mathbf{H}[n-1] \\ \mathbf{h} \end{bmatrix} \right)^{-1} \cdot \left(\begin{bmatrix} \mathbf{H}[n-1] & \mathbf{h} \end{bmatrix} \begin{bmatrix} \mathbf{X}[n-1] \\ X[n] \end{bmatrix} \right) \\ &= (\mathbf{H}^T[n-1]\mathbf{H}[n-1] + \mathbf{h}[n]\mathbf{h}^T[n])^{-1} \cdot (\mathbf{H}^T[n-1]\mathbf{X}[n-1] + \mathbf{h}[n]X[n]) \end{aligned}$$

definiendo $\mathbf{C}[n] = (\mathbf{H}^T[n]\mathbf{H}[n])^{-1}$ y utilizando la indentidad de Woodbury [16] el primer factor de la ecuación anterior se reescribe como:

$$\begin{aligned}
 \mathbf{C}[n] &= (\mathbf{C}[n-1]^{-1} + \mathbf{h}[n]\mathbf{h}^T[n])^{-1} \\
 (2.77) \quad &= \mathbf{C}[n-1] - \left(\frac{\mathbf{C}[n-1]\mathbf{h}[n]\mathbf{h}^T[n]\mathbf{C}[n-1]}{1 + \mathbf{h}^T[n]\mathbf{C}[n-1]\mathbf{h}[n]} \right) \\
 &= (\mathbf{I} - \mathbf{K}[n]\mathbf{h}^T[n]) \mathbf{C}[n-1]
 \end{aligned}$$

con

$$(2.78) \quad \mathbf{K}[n] = \frac{\mathbf{C}[n-1]\mathbf{h}[n]}{1 + \mathbf{h}^T[n]\mathbf{C}[n-1]\mathbf{h}[n]}$$

reemplazando 2.77 en 2.76 se obtiene:

$$\begin{aligned}
 \hat{\theta}[n] &= (\mathbf{I} - \mathbf{K}[n]\mathbf{h}^T[n]) \mathbf{C}[n-1] \cdot (\mathbf{H}^T[n-1]\mathbf{X}[n-1] + \mathbf{h}[n]X[n]) \\
 (2.79) \quad &= (\mathbf{I} - \mathbf{K}[n]\mathbf{h}^T[n]) \mathbf{C}[n-1]\mathbf{H}^T[n-1]\mathbf{X}[n-1] + (\mathbf{I} - \mathbf{K}[n]\mathbf{h}^T[n]) \mathbf{C}[n-1]\mathbf{h}[n]X[n] \\
 &= \mathbf{C}[n-1]\mathbf{H}^T[n-1]\mathbf{X}[n-1] - \mathbf{K}[n]\mathbf{h}^T[n]\mathbf{C}[n-1]\mathbf{H}^T[n-1]\mathbf{X}[n-1] + \\
 &\quad (\mathbf{C}[n-1]\mathbf{h}[n] - \mathbf{K}[n]\mathbf{h}^T[n]\mathbf{C}[n-1]\mathbf{h}[n]) X[n]
 \end{aligned}$$

considerando los valores:

$$(2.80) \quad \hat{\theta}[n-1] = \mathbf{C}[n-1]\mathbf{H}^T[n-1]\mathbf{X}[n-1]$$

$$(2.81) \quad \mathbf{C}[n-1]\mathbf{h}[n] - \mathbf{K}[n]\mathbf{h}^T[n]\mathbf{C}[n-1]\mathbf{h}[n] = \mathbf{K}[n]$$

y reemplazando en la ecuación 2.79, se obtiene

$$\begin{aligned}
 \hat{\theta}[n] &= \hat{\theta}[n-1] - \mathbf{K}[n]\mathbf{h}^T[n]\hat{\theta}[n-1] + \mathbf{K}[n]X[n] \\
 (2.82) \quad &= \hat{\theta}[n-1] + \mathbf{K}[n] \left(X[n] - \mathbf{h}^T[n]\hat{\theta}[n-1] \right)
 \end{aligned}$$

Finalmente, con las ecuaciones 2.77, 2.78 y 2.82, es posible crear un algoritmo secuencial para el estimador $\hat{\theta}[n]$ donde su matriz de covarianza será:

$$(2.83) \quad \mathbf{C}_{\hat{\theta}[n]} = \mathbf{C}[n]\sigma^2$$

2.6. Información del estado del canal

La “información del estado del canal” (CSI) se refiere al conocimiento de ciertas propiedades del canal que afectan la señal enviada. Esta información se estima en el receptor y usualmente es enviada al emisor que adapta la señal de acuerdo a esta información. La adaptación, en base al conocimiento de canal, puede producir mejoras significativas en el desempeño general del esquema de comunicaciones [19], donde las más utilizadas son la adaptación de potencia y codificación.

Existen dos clases de “información del estado del canal”: la instantánea o de tiempo corto, y la estadística o de tiempo largo. La explicación del significado de cada uno de estos conceptos se realizará asumiendo un canal inalámbrico como el descrito en la sección 2.3, ecuación 2.33.

Información Instantánea

Para la “información instantánea del estado del canal” es necesario conocer exactamente el valor de las constantes $h_\ell[n]$ para todo instante de tiempo. Esta información da la oportunidad de adaptar la señal transmitida según la respuesta al impulso que tenga el canal. La información instantánea del canal es variable y cambiante en el tiempo por lo que para obtenerla, se debe estimar cada cierto tiempo en el receptor.

Información Estadística

Para la “información estadística del canal” es necesario conocer las características estadísticas de la variable aleatoria $h_\ell[n]$. Por ejemplo algunas características que pudieran ser de interés son la matriz de correlación de $h_\ell[n]$ la ganancia del canal o la potencia de la línea de visión. También es de utilidad obtener parámetros estadísticos sobre la función de distribución de h_ℓ .

2.6.1. Método de secuencias piloto

La estimación de la respuesta al impulso del canal se hace bajo el supuesto que las señales de entrada y salida del canal son conocidas para los tiempos en los cuales se realiza la estimación. Es posible considerar que el estado del canal se mantendrá constante durante un espacio de tiempo determinado, que es el tiempo de coherencia. Si el tiempo coherencia es lo suficientemente largo, en relación a la frecuencia de muestreo, se podrá estimar el canal utilizando solo una fracción del total del mensaje. El método de las secuencias piloto, o secuencias de entrenamiento fija una secuencia de bits constante de largo determinado que se enviarán periódicamente con el objetivo de obtener la caracterización del canal. La figura 2.13 muestra el funcionamiento del método de las secuencias de entrenamiento:

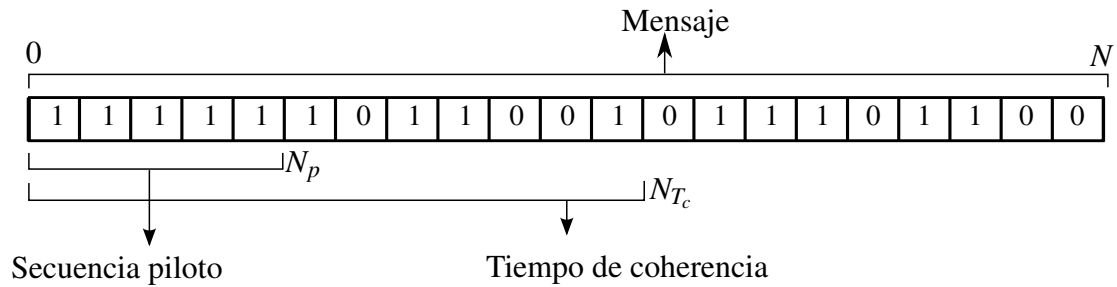


Figura 2.13: Secuencias piloto

donde N_p representa el largo de la secuencia piloto a utilizar, N_{T_c} el largo de la secuencia de bits transmitidos durante el tiempo de coherencia y N el largo total del mensaje. El largo y periodo de una secuencia piloto queda determinado por el tiempo de coherencia y el estimador que se utiliza, en general, si el tiempo de coherencia es menor al tiempo mínimo necesario para estimar las características del canal, entonces es posible obtener, solo, la información estadística del canal. Es necesario tener una relación adecuada entre el N_{T_c} y N_p , ya que si estos números son parecidos, se estará utilizando todo el mensaje solo para estimar el canal, lo cual no tiene sentido si se desea realizar un control “online” de potencia o de codificación.

2.7. Estado del arte

La estimación estadística de las características del canal es una técnica muy utilizada en esquemas inalámbricos actuales, como WIFI, Bluetooth, DSB y en general cualquier esquema de comunicación que use el standar IEEE 802.11p [14]. Sin embargo su uso en satélites tipo “cubesat” no ha sido muy extendido. Esto se refleja en que las misiones que utilizan el standar “cubesat” generalmente no llevan un esquema de estimación del canal [1] [18].

Es posible adaptar los estimadores utilizados para otras tecnologías inalámbricas al standar “cubesat”. Para analizar la factibilidad de esto es importante escoger los estimadores y probarlos en simulaciones con características similares a las esperadas para el esquema real. Los estimadores, lineal y secuencial presentados en las secciones 2.5.3 y 2.5.5, fueron escogidos por su facilidad de implementación y su amplia utilización en otras comunicaciones inalámbricas [9] [3] [15] [26] [32]. A su vez se designaron las características particulares esperadas para el canal real del proyecto SUCHAI, como la dispersión de Doppler (sección 2.3.6), la tasa de datos y la relación señal a ruido (sección 2.4.4). Otro factor a tener en consideración es que la modulación de los esquemas que usan, habitualmente, la estimación de las características del canal ocupan múltiples subcarriers multiplexados (OFDM por sus siglas en inglés “orthogonal frequency-division multiplexing”) lo cual deja la posibilidad de definir algunos subcarriers solo para estimación del canal, por lo que la implementación y factibilidad de los estimadores aplicados a un esquema FSK debe ser analizada.

Capítulo 3

Implementación

3.1. Modelo del canal

El modelo escogido para representar el canal de comunicaciones satélite-estación terrena es el discreto en banda base con ruido aditivo Gaussiano presentado en la sección 2.3. En la figura 3.1 se muestra un esquema de la estructura del modelo. La señal $x[n]$ es el equivalente en banda base discreto de la señal modulada en FSK que se desea enviar desde el emisor hasta el receptor y la señal $y[n]$ es el equivalente discreto en banda base de la onda obtenida en la antena receptora. Los “taps”, $h_\ell[n]$, son la representación del canal físico por el que viaja la señal. A esta representación se le conoce como “tap delay line”, o **TDL** por sus siglas en ingles y es un modelo ampliamente utilizado [26] [4] [32] para canales inalámbricos. $\omega[n]$ es un proceso aleatorio Gaussiano blanco de simetría circular, donde $\omega[n] \sim \mathcal{CN}(0, N_0/2)$.

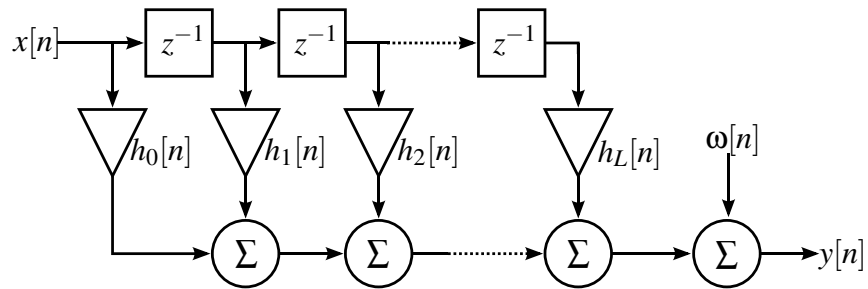


Figura 3.1: Esquema del canal TDL

$$(3.1) \quad y[n] = \sum_{\ell=0}^L h_\ell[n]x[n - \ell] + \omega[n]$$

En la figura 3.2 se muestra un esquema completo del modelo utilizado para representar el canal de comunicaciones. Tx son los datos enviados desde la fuente. Rx son los datos recibidos por el receptor. El bloque **Modulador** obtiene el equivalente en banda base discreto de la señal modulada en FSK con los datos Tx y el bloque **Demodulador** hace el proceso inverso.

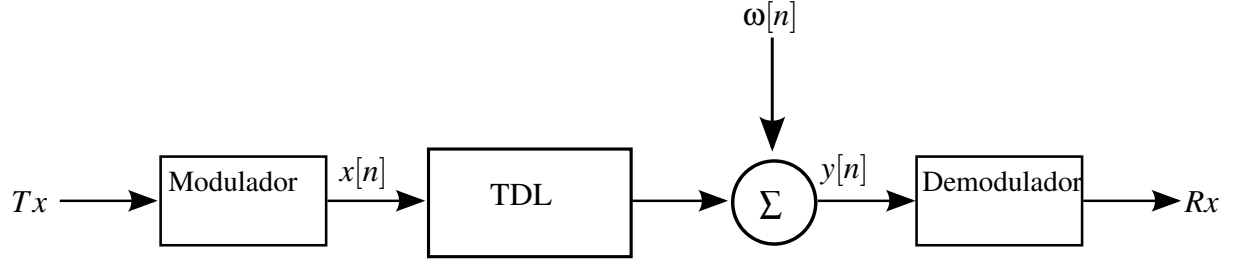


Figura 3.2: Esquema del canal completo

Los valores $x[n]$ y $y[n]$ son conocidos para todo instante n debido a que pueden obtenerse en la antena emisora y receptora respectivamente.

3.2. Estimadores Implementados

Se desarrollaron metodologías para caracterizar los estados h_ℓ del canal, utilizando estimación estadística de señales. Los estimadores estadísticos utilizados se explican a continuación.

3.2.1. Estimador lineal

Para este estimador se asume que los coeficientes $h_\ell[n]$ a estimar se mantienen constantes durante el tiempo de coherencia del canal. El supuesto es válido debido a que durante este tiempo los coeficientes no sufren cambios notables. Siguiendo este razonamiento el modelo del canal se puede reescribir:

$$(3.2) \quad y[n] = \sum_{\ell=0}^L h_{\ell,c} x[n-\ell] + \omega[n] \quad cT_c \leq nT_s < (c+1)T_c$$

donde c es un número entero mayor o igual a cero, que representa las muestras pertenecientes a un intervalo $[t, t + T_c]$ en que los coeficientes $h_{\ell,c}$ son constantes. Definiendo los vectores $\mathbf{h}_c^T = [h_{0,c} \ h_{1,c} \ \dots \ h_{L,c}]$, $\mathbf{x}^T[n] = [x[n] \ x[n-1] \ \dots \ x[n-L]]$ y el valor entero $M = \lfloor T_c/T_s \rfloor$ donde $\lfloor \cdot \rfloor$ es la función “cajón inferior”. Es posible escribir la ecuación 3.2 de forma más compacta:

$$(3.3) \quad y[n] = \mathbf{x}^T[n] \cdot \mathbf{h}_c + \omega[n] \quad cM \leq n < (c+1)M$$

para cada espacio de tiempo c el vector \mathbf{h}_c tendrá valores constantes, y por lo tanto será posible representar la ecuación del canal de forma matricial para cada intervalo de tiempo c por separado

$$(3.4) \quad \mathbf{Y}_c = \mathbf{X}_c \cdot \mathbf{h}_c + \boldsymbol{\omega}$$

(3.5)

$$\mathbf{Y}_c = \begin{bmatrix} y[cM] \\ y[cM+1] \\ \vdots \\ y[(c+1)M-1] \end{bmatrix} \rightarrow (M \times 1)$$

$$\mathbf{X}_c = \begin{bmatrix} x[cM] & x[cM-1] & \cdots & x[cM-L] \\ x[cM+1] & x[cM] & \cdots & x[cM+1-L] \\ \vdots & \ddots & & \vdots \\ x[(c+1)M-1] & x[(c+1)M-2] & \cdots & x[(c+1)M-(L+1)] \end{bmatrix} \rightarrow (M \times L+1)$$

$$\boldsymbol{\omega} = \begin{bmatrix} \omega[cM] \\ \omega[cM+1] \\ \vdots \\ \omega[(c+1)M-1] \end{bmatrix} \rightarrow (M \times 1)$$

Las matrices \mathbf{Y}_c y \mathbf{X}_c son conocidas para todo c ya que se construyen a través de las señales $y[n]$ y $x[n]$. Para modelos lineales como el descrito en la ecuación 3.4 es posible plantear un estimador óptimo $\hat{\mathbf{h}}_c$ que logra la cota de Cramer-Rao (CRLB):

$$(3.6) \quad \hat{\mathbf{h}}_c = (\mathbf{X}_c^T \cdot \mathbf{X}_c)^{-1} \mathbf{X}_c^T \mathbf{Y}_c$$

donde la matriz de covarianza teórica del estimador es $\mathbf{C}_{\hat{\mathbf{h}}_c} = (\mathbf{X}_c^T \cdot \mathbf{X}_c)^{-1} N_0$.

El tiempo de coherencia se relaciona directamente con la dispersión de Doppler D_s mediante la ecuación 3.7

$$(3.7) \quad T_c \approx 1/(4D_s)$$

si este valor es significativamente mayor a la frecuencia de muestreo, entonces se cumplirá $M \gg L$ y se tendrán suficientes muestras para calcular el estimador $\hat{\mathbf{h}}_c$. Debido a que se espera que la dispersión de Doppler para el canal del proyecto sea baja, debido a que la antena terrestre se encuentra estática (como se mencionó en la sección 2.3.6), se contará con muestras abundantes para calcular el estimador.

Algoritmo de estimación

Para realizar la estimación del canal objetivo es necesario procesar las señales de entrada y salida del canal ($x[n]$ e $y[n]$) y calcular el estimador $\hat{\mathbf{h}}_c$, con la ecuación 3.6, para cada intervalo de

tiempo c . Para esto es necesario contar con un algoritmo que realice estas operaciones en cada ciclo c . En la figura 3.5 se muestra el algoritmo general que se utilizó para obtener el estimador lineal $\hat{\mathbf{h}}_c$.

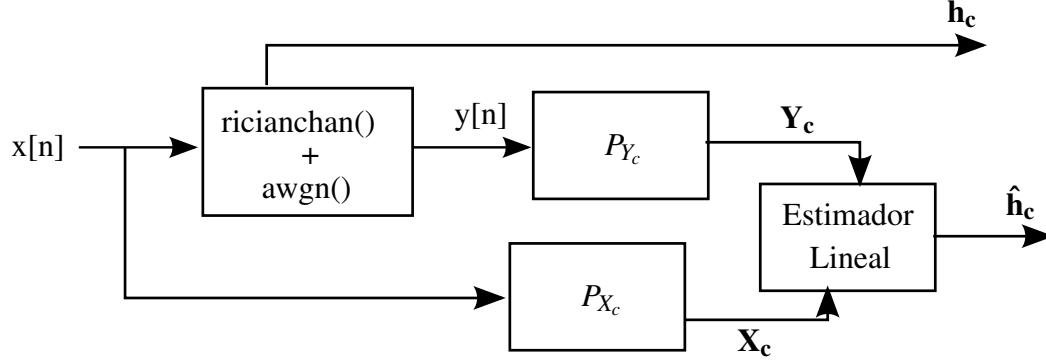


Figura 3.3: Algoritmo de estimación lineal

El bloque P_{Y_c} corresponde a una serie de M registros que forman la matriz \mathbf{Y}_c para cada espacio de tiempo c y el bloque P_{X_c} es un bloque más complejo que a través de M registros y reordenación de datos construye la matriz \mathbf{X}_c . Con éstas dos matrices se estima $\hat{\mathbf{h}}_c$ en el bloque “Estimador Lineal” para todo espacio de tiempo c .

3.2.2. Estimador lineal secuencial

Es posible plantear un algoritmo secuencial para calcular el estimador lineal de la ecuación 3.6. Se definen las matrices $\mathbf{X}_c[n]$ y $\mathbf{Y}_c[n]$ tal que:

$$(3.8) \quad \mathbf{X}_c[n] = \begin{bmatrix} \mathbf{X}_c[n-1] \\ \mathbf{x}^T[n] \end{bmatrix} \rightarrow \begin{bmatrix} (n - cM + 1) \times L \\ 1 \times L \end{bmatrix}$$

$$(3.9) \quad \mathbf{X}_c[cM] = \mathbf{x}^T[cM] \quad cM \leq n < (c+1)M$$

$$\mathbf{Y}_c[n] = [y[cM] \ \dots \ y[n-1] \ y[n]]^T \rightarrow ((n - cM + 1) \times 1)$$

se observa que para $n_c = (c+1)M - 1$ se cumplirá que $\mathbf{X}_c[n_c] = \mathbf{X}_c$ y $\mathbf{Y}_c[n_c] = \mathbf{Y}_c$. Con esta misma lógica es posible definir el estimador $\hat{\mathbf{h}}_c[n]$:

$$(3.10) \quad \hat{\mathbf{h}}_c[n] = (\mathbf{X}_c^T[n] \cdot \mathbf{X}_c[n])^{-1} \mathbf{X}_c^T[n] \mathbf{Y}_c[n] \quad cM \leq n < (c+1)M$$

$$\hat{\mathbf{h}}_c[n_c] = \hat{\mathbf{h}}_c$$

En la figura 3.4 se muestra el algoritmo secuencial que se ocupa para calcular el estimador $\hat{\mathbf{h}}_c[n]$ para todo tiempo n :

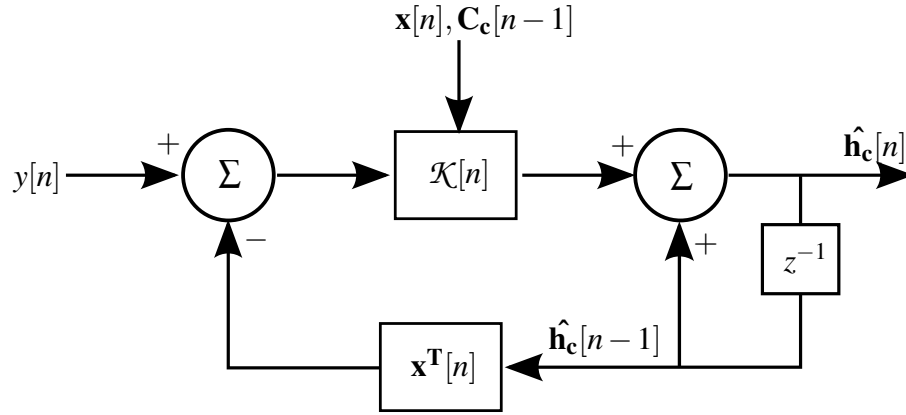


Figura 3.4: Diagrama de bloques que representa el algoritmo secuencial

$$(3.11) \quad \hat{\mathbf{h}}_{\mathbf{c}}[n] = \hat{\mathbf{h}}_{\mathbf{c}}[n-1] + \mathcal{K}_{\mathbf{c}}[n] (y[n] - \mathbf{x}^T[n] \hat{\mathbf{h}}_{\mathbf{c}}[n-1])$$

$$(3.12) \quad \mathcal{K}_{\mathbf{c}}[n] = \frac{\mathbf{C}_{\mathbf{c}}[n-1] \mathbf{x}[n]}{1 + \mathbf{x}^T[n] \mathbf{C}_{\mathbf{c}}[n-1] \mathbf{x}[n]}$$

$$(3.13) \quad \mathbf{C}_{\mathbf{c}}[n] = (\mathbf{I} - \mathcal{K}_{\mathbf{c}}[n] \mathbf{x}^T[n]) \mathbf{C}_{\mathbf{c}}[n-1]$$

Es necesario definir valores iniciales para cada espacio de tiempo c de manera que exista consistencia en el algoritmo. Por esta razón se definen:

$$(3.14) \quad \begin{aligned} \hat{\mathbf{h}}_{\mathbf{c}}[cM-1] &= \hat{\mathbf{h}}_{\mathbf{c}-1}[cM-1], \quad \hat{\mathbf{h}}_{\mathbf{c}}[-1] = \mathbf{0}_{(L,1)} \\ \mathbf{C}_{\mathbf{c}}[cM-1] &= (\mathbf{C}_0)_{(L,L)} \end{aligned} \quad \forall c \in \mathbb{N}_0$$

donde $C_{L,L}$ es una matriz cuadrada de $(L \times L)$ elementos que contiene las condiciones iniciales en su diagonal y $0_{(L,1)}$ una matriz de $(L \times 1)$ elementos iguales a 1.

Algoritmo de estimación

El algoritmo de estimación para el estimador secuencial es similar al lineal, con la diferencia que es necesario menor procesamiento para obtener las matrices en cada ciclo. La figura 3.5 muestra el esquema del algoritmo.

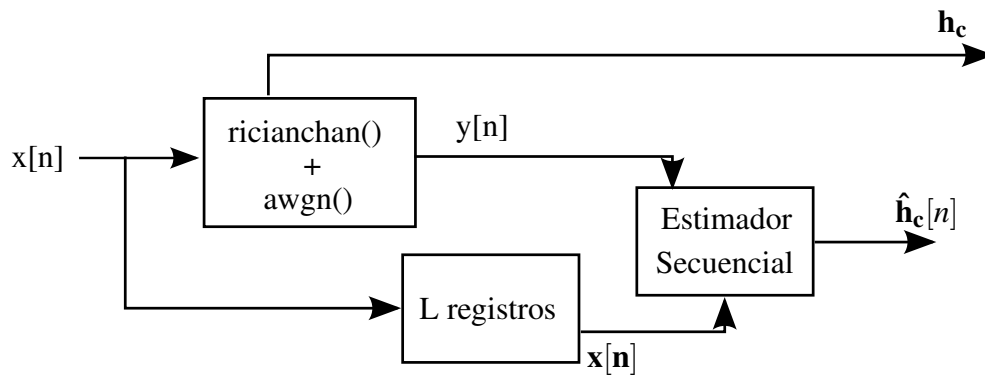


Figura 3.5: Algoritmo de estimación secuencial lineal

Se observa que no es necesario realizar procesamiento a la señal $y[n]$ y para la señal $x[n]$ lo único necesario son L registros que mantengan el valor $x[n]$ en instantes anteriores. Luego, con estos registros, es posible construir la matriz $\mathbf{x}[n]$ y realizar el algoritmo secuencial presentado en la figura 3.4. En el caso de este algoritmo de estimación el procesamiento se realiza directamente en cada ciclo n a diferencia del lineal que debe esperar M muestras para realizar la estimación. Esta diferencia se refleja en una mayor facilidad de implementación por “hardware” y mayor rapidez de proceso cuando se comparan circuitos ad-hoc.

3.3. Simulación del canal físico

Para probar el desempeño de los estimadores se realizaron simulaciones computacionales en Matlab[®], emulando el comportamiento de un canal real. Para procesar las simulaciones se ocupó un computador personal con un procesador Intel[®] i7-2620M con 4 núcleos de 2.7GHz y 3 GB de RAM con un sistema operativo Linux Mint 12 de 32 bits corriendo la versión R2010a de Matlab[®]. Para simular cada bloque del esquema de comunicación se utilizaron funciones nativas de Matlab[®] pertenecientes al toolbox de comunicaciones. El diagrama de bloques de la simulación básica se muestra en la figura 3.6.

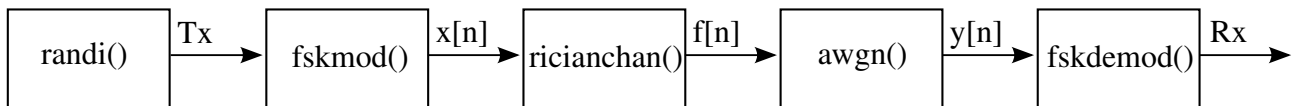


Figura 3.6: Esquema de la simulación del canal

Cada bloque es una función de Matlab con parámetros específicos que determinan el comportamiento de la simulación. La selección de las características de cada bloque se realizó de forma que el comportamiento de la simulación cumpliera los requerimientos conocidos del esquema de comunicaciones del proyecto [SUCHAI](#) presentado en la sección 2.4.3. Los valores y la función de cada bloque se presentan a continuación.

funcionamiento del modulador, \mathbf{M} es el número de símbolos que tiene el modulador, **freqsep** es la diferencia entre la frecuencia máxima y mínima que se utilizará, **nsamp** es el número de muestras por símbolo que contendrá la señal de salida y **Fs** es la frecuencia de muestreo en la salida. Los valores que se han fijado para la simulación se encuentran en la tabla 3.2.

Función	Parámetros	Valor
fskmod (Tx,M,freqsep,nsamp,Fs)	M	2
	freqsep	10 [KHz]
	nsamp	16
	Fs	20 [KHz]

Tabla 3.2: Parámetros de función fskmod()

Se utilizó FSK debido a que el esquema de comunicaciones del proyecto [SUCHAI](#) ocupa esta modulación. Los parámetros de este bloque fueron elegidos de forma tal que el modulador de la simulación se pareciera al modulador del proyecto, mostrado en la sección 2.4.4. La frecuencia de separación se fijó en el valor máximo permitido por el esquema del proyecto (10[KHz]). La frecuencia de muestreo se estableció tal que $\mathbf{Fs}=2 \cdot \mathbf{freqsep}$ debido a que la frecuencia máxima alcanzada por la señal en banda base será menor o igual a **freqsep** y por lo tanto, por el teorema del muestreo, es posible reconstruir perfectamente la señal. El valor **nsamp** se obtiene teniendo en cuenta que la tasa de transferencia de datos, $f_b[\text{bits}/\text{segundo}]$, puede ser calculada con los valores **Fs** y **nsamp**:

$$(3.15) \quad f_b = \frac{nbits}{T_{sim}} = \frac{nbits}{nsamp \cdot T_s \cdot nbits} = \frac{F_s}{nsamp}$$

donde *nbits* son la cantidad de bits enviados en un tiempo de simulación T_{sim} . La tasa de transferencia del proyecto [SUCHAI](#) es de $1200[\text{bits}/\text{s}]$. Ocupando la relación de la ecuación 3.15 es posible calcular la tasa de la simulación $f_b = F_s/nsamp = 20000/16 = 1250[\text{bits}/\text{s}]$ similar a la ocupada en el proyecto.

Bloque ricianchan()

La función **ricianchan(Ts,dop,K,τ^T,adb^T)** construye un filtro **TDL** en donde al menos la magnitud de un parámetro $h_l[n]$ es una variable aleatoria que cumple con la distribución de Rice vista en la sección 2.3.7.

$$(3.16) \quad h_\ell[m] = \sqrt{\frac{K}{K+1}} \sigma_\ell e^{j\theta} + \sqrt{\frac{1}{K+1}} C\mathcal{N}(0, \sigma_\ell^2)$$

Los parámetros de la función **ricianchan()** definen el comportamiento de las variables aleatorias que forman el filtro. La variable **Ts** es el período de muestreo, que se define como $T_s = 1/F_s$ y

F_s fue fijado anteriormente para el bloque modulador. La variable **dop** constituye la dispersión de Doppler del canal que influye directamente con el tiempo de coherencia del mismo. Esta variable se fijó en 10 [Hz] que constituye un valor mayor del esperado para canales satelitales del tipo del proyecto (visto en la sección 2.3.6). El valor **K** es la constante de mismo nombre en la función de distribución Riciana que representa la proporción entre la potencia de la componente **LOS** y los demás caminos. Se definió **K** = 9 debido a que en el canal satelital del proyecto la línea de visión siempre se encuentra presente y se prevé que el aporte de los múltiples caminos que puedan haber no serán preponderantes debido a los pocos y pequeños obstáculos que se encuentran cercanos a la estación terrena. El parámetro Ω , visto en la sección 2.3.7, es normalizado a 1 por defecto en Matlab.

Los vectores **adb** y τ se definen tal que:

$$(3.17) \quad \mathbf{adb} = \begin{bmatrix} \mathcal{P}(a_1^b)[dB] \\ \mathcal{P}(a_2^b)[dB] \\ \vdots \\ \mathcal{P}(a_P^b)[dB] \end{bmatrix} \quad \tau = \begin{bmatrix} \tau_1[s] \\ \tau_2[s] \\ \vdots \\ \tau_P[s] \end{bmatrix}$$

$$(3.18) \quad h_\ell[n] = \sum_{i=1}^P a_i^b(n/F_s) \text{sinc}[\ell - \tau_i(n/F_s)F_s]$$

donde $\mathcal{P}(a_i^b)$ es la potencia en [dB] de las variables $a_\ell[n]$ y los valores τ_i representa los retardos, en segundos, de cada camino i . Estos valores se fijaron $\mathbf{adb}^T = [0 \ -100 \ -100 \ -100][dB]$ y $\tau^T = [0 \ 10^{-8} \ 10^{-7} \ 10^{-6}][s]$. Los valores fijados para todas las variables se resumen en la tabla 3.3.

Función	Parámetros	Valor
ricianchan ($T_s, \text{dop}, \mathbf{K}, \tau^T, \mathbf{a}^T$)	T_s	$1/F_s = 50[\mu s]$
	dop	10 [Hz]
	K	9
	τ^T	$[0 \ 10^{-8} \ 10^{-7} \ 10^{-6}]$
	adb ^T	$[0 \ -100 \ -100 \ -100]$
Parámetros Derivados	Ω	1
	$v^2 = K\Omega/(K+1)$	0.9
	$\sigma^2 = \Omega/(2(1+K))$	0.05
	$T_c = 1/(4\text{dop})$	25[ms]

Tabla 3.3: Parámetros de función **ricianchan**()

Es posible recuperar los valores exactos de las atenuaciones $\mathbf{a}[\mathbf{n}]^T = [a_1^b[n] \dots a_P^b[n]]$ para cada realización específica de la simulación. Con los vectores $\mathbf{a}[\mathbf{n}]$ y τ es posible calcular, a través de la ecuación 3.18, el vector $\mathbf{h}[\mathbf{n}]^T = [h_1[n] \dots h_L[n]]$ que corresponde a la respuesta al impulso del filtro **TDL**.

En la figura 3.8 se muestra el histograma del módulo de $|h_0[n]|$ para una realización de la simulación en comparación con la distribución teórica de Rice con los mismos parámetros anteriores.

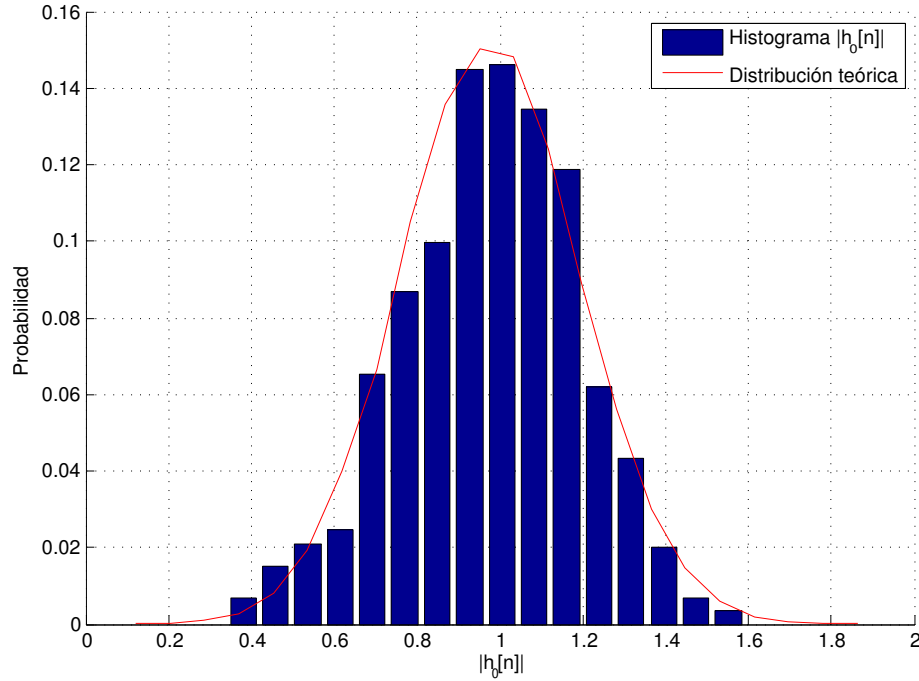


Figura 3.8: Histograma de $|h_0[n]|$. La simulación se realizó utilizando el canal Riciano de matlab con los parámetros mencionados en la tabla 3.3. Se procesaron 16.000 muestras aleatorias, obteniéndose un valor central de 1,0080 y desviación estándar de 0,2130. La simulación se encuentra en el script rician.mat

Bloque awgn()

La función `rxSig = awgn(f[n], SNR[db])` agrega ruido blanco Gaussiano a la señal $\mathbf{f}[n]$ obtenida a la salida del canal de Rice. La variable `SNR[db]` representa la relación señal a ruido en decibeles entre la señal $\mathbf{f}[n]$ y el ruido $\omega[n]$. Considerando que la señal $\mathbf{f}[n]$ está normalizada a una potencia de 0 db se tiene:

$$y[n] = f[n] + \omega[n] \quad \omega[n] \sim \mathcal{CN}(0, N_0/2)$$

$$(3.19) \quad SNR[db] = 10 \log \left(\frac{P(f[n])}{P(\omega[n])} \right) = P(f[n])[db] - P(\omega[n])[db] = -N_0[db]$$

$$SNR[u] = \frac{1}{N_0}$$

considerando que la varianza de la variable aleatoria $\omega[n]$ es N_0 se obtiene una relación directa entre la variable **SNR** y la varianza del ruido blanco aleatorio:

$$(3.20) \quad \text{var}(\omega[n]) = \sigma_{\omega}^2 = \frac{1}{\text{SNR}}.$$

En la simulación, se utiliza la variable E_b/N_0 para medir el valor del ruido. La conversión entre E_b/N_0 y **SNR** es: $\text{SNR}[dB] = E_b/N_0[db] + 10\log\left(\frac{f_b}{F_s}\right)$ donde f_b es la tasa de transferencia de datos. En el caso de la simulación se cumple la relación de la ecuación 3.21:

$$(3.21) \quad \text{SNR}[db] = E_b/N_0[db] - 10\log(\text{nsamp})$$

Los valores utilizados en la simulación son un barrido en la variable E_b/N_0 , tal que: $E_b/N_0 = [0 \dots 20]$, que es igual a un barrido en $\text{SNR}[dB] = [-12 \dots 8][dB]$.

Bloque **fskdemod()**

La función **fskmod** (**y[n],M,freqsep,nsamp,FS**) demodula la señal **y[n]** y entrega el mensaje obtenido por el receptor. Los parámetros (**M, freqsep,nsamp,FS**) son iguales a los fijados en el bloque **fskmod**. En la figura 3.9 se muestra el comportamiento de los bloques **fskmod()** y **fskdemod()** para el caso básico en donde solo se tiene un canal ruidoso **AWG**.

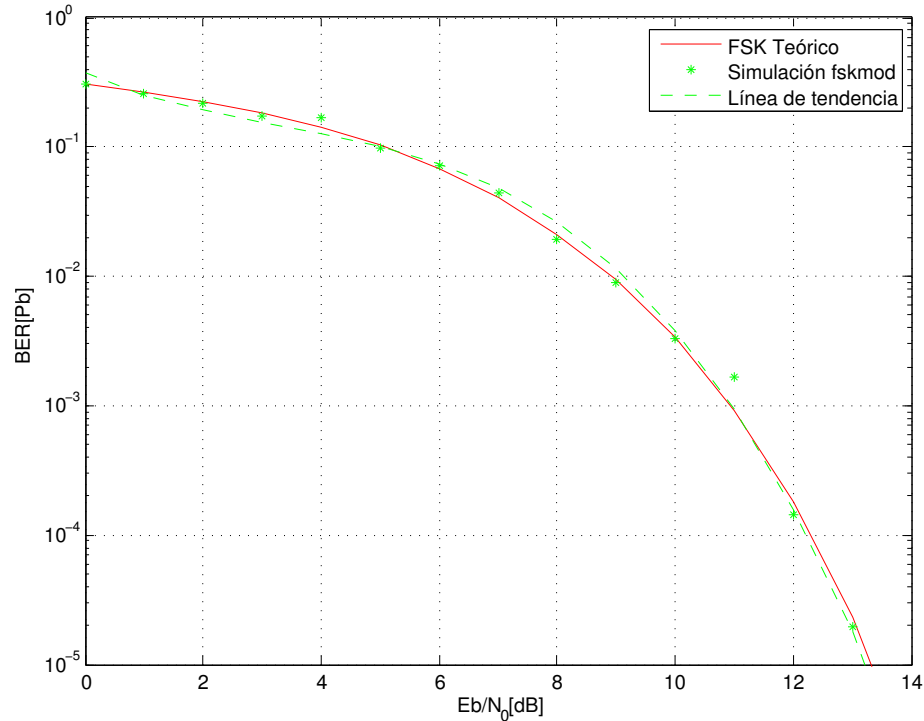


Figura 3.9: Comportamiento del modulador. La línea continua representa el BER teórico de una modulación FSK no coherente binaria, la línea segmentada son los valores medios obtenidos por los bloques `fkmod()` y `fskdemod()` para diferentes valores de E_b/N_0 producidos por el bloque `awgn()` y la línea de tendencia es la función exponencial que mejor se ajusta a los datos. La simulación se hizo con la herramienta “bertool” de Matlab que realiza un test de montecarlo hasta llegar a 100 errores o 10^8 bits para cada E_b/N_0 . el código de la simulación se encuentra en el archivo `fsksim.m`

3.3.1. Comportamiento de la simulación

Al agregar el canal Riciano a la simulación, se obtiene el comportamiento mostrado en la figura 3.10. Se observa que el rendimiento del canal es peor que el del modulador FSK teórico con un canal AWG.

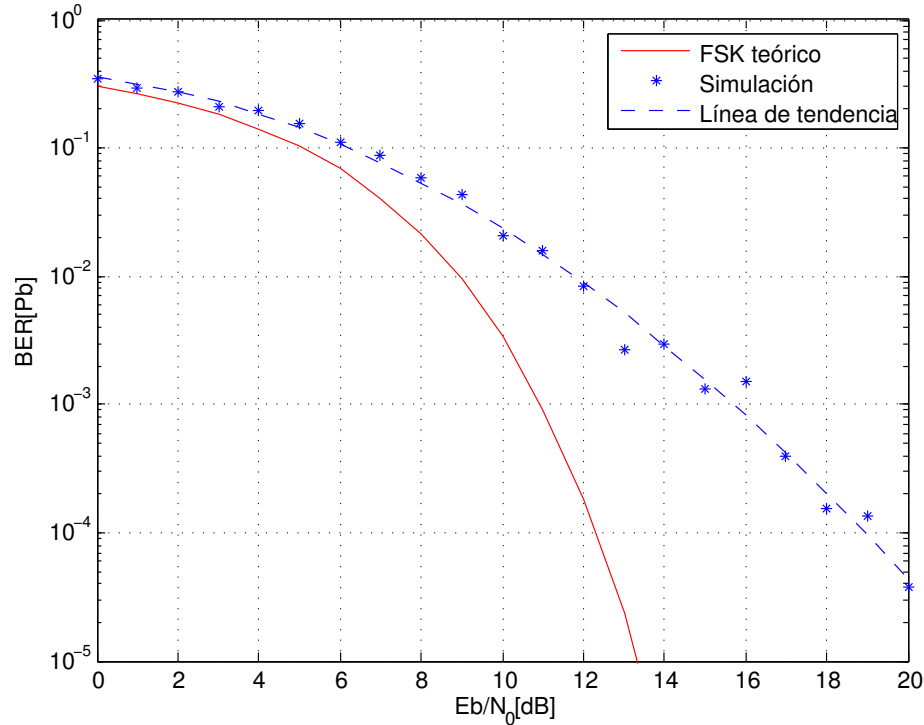


Figura 3.10: Comportamiento de la simulación con canal Riciano y ruido AWG. La línea continua representa el comportamiento teórico de un modulador FSK no-coherente, los datos de la simulación se obtuvieron utilizando el método de Montecarlo hasta llegar a 100 errores o procesar 10^8 bits y la línea de tendencia es la función exponencial que mejor se ajusta a los datos. El código de la simulación se encuentra en el archivo riciansim.m

3.4. Procedimiento de simulación

Se realizaron simulaciones para probar el comportamiento de los estimadores. En particular se realizaron procedimientos para medir la varianza media del estimador y la diferencia entre el valor estimado medio y el valor real del canal. También se hicieron simulaciones para obtener la información estadística del canal como las constantes K y Ω de la distribución Riciana o la correlación en el tiempo de los parámetros $h_\ell[n]$.

3.4.1. Procedimiento para obtener las características del estimador

Las salidas $y[n]$ se realizaron fijando las constantes $h_\ell[n]$ de un canal aleatorio Riciano creado por Matlab con las cualidades vistas en la sección 3.3 y agregando ruido AWG de distintas intensidades. La entrada $x[n]$ se realizó modulando en FSK un mensaje aleatorio Tx que permanece constante para toda la simulación. Se hicieron 100 realizaciones para cada intensidad de ruido en

estudio. Las intensidades de ruido que se estudiaron fueron los valores de E_b/N_0 entre 0[dB] y 20[dB]. En la figura 3.11 se muestra el esquema de la realización de la simulación.

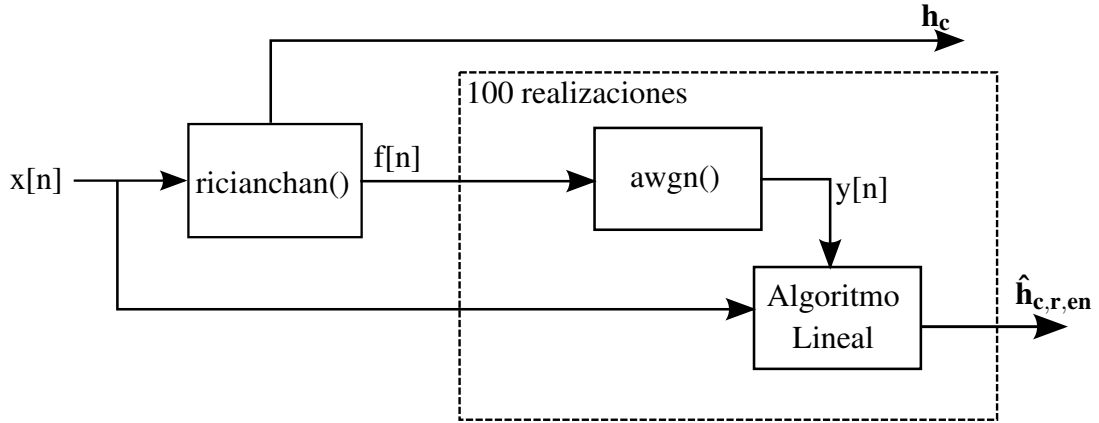


Figura 3.11: Procedimiento de simulación para estimación lineal

El bloque “Algoritmo Lineal” realiza el algoritmo de estimación dependiendo del estimador en estudio y `awgn()` determina el nivel de ruido de la simulación. En la salida de cada realización se obtiene un set de estimadores $\hat{\mathbf{h}}_{c,r,en}$ para el estimador lineal y $\hat{\mathbf{h}}_{c,r,en}[n]$ para el estimador lineal secuencial donde los sub índices r, en y c denotan que el estimador pertenece a la realización r y se realizó con un $E_b/N_0 = en$ en un espacio de tiempo c . Luego con estos valores es posible calcular la media y la varianza del estimador.

3.4.2. Procedimiento para obtener las características del canal

El procedimiento para obtener las características del canal es diferente del que se utilizó para obtener las características del estimador. La diferencia es que no se deben fijar los parámetros $h_\ell[n]$ del canal ya que se busca obtener parámetros estadísticos del comportamiento aleatorio de esta variable. Salvo por esta diferencia, el algoritmo de simulación realizado es igual al anterior. Se realizaron 100 realizaciones del algoritmo y se obtienen los mismos estimadores $\hat{\mathbf{h}}_{c,r,en}$ para el estimador lineal y $\hat{\mathbf{h}}_{c,r,en}[n]$ para el estimador lineal secuencial.

3.5. Secuencias piloto

Se realizaron las simulaciones nombradas anteriormente para diferentes largos de secuencias piloto y se hicieron comparaciones entre los comportamientos de cada estimador frente al cambio en el largo de la secuencia que se utiliza para realizar la estimación. Las secuencias utilizadas se muestran en la tabla 3.4.

Secuencia	Largo de la secuencia [Muestras]
1	$M = 500$
2	$0.5M = 250$
3	$0.25M = 125$
4	$0.1M = 50$
5	$0.05M = 25$
6	$0.01M = 5$

Tabla 3.4: Parámetros de las secuencias piloto

Capítulo 4

Discusión de resultados

4.1. Resultados estimador lineal

4.1.1. Comportamiento del estimador

Los resultados para analizar el comportamiento del estimador lineal se obtuvieron utilizando la metodología expuesta en la sección 3.4.1, obteniéndose de esta forma un set de valores $\hat{\mathbf{h}}_{\mathbf{c},r,\text{en}}$. Es posible obtener una aproximación de la media y la varianza media del estimador realizando operaciones sobre estos valores:

$$(4.1) \quad E(\hat{\mathbf{h}}_{\mathbf{c},\text{en}}) \approx \frac{1}{R} \cdot \sum_r \hat{\mathbf{h}}_{\mathbf{c},r,\text{en}}$$

$$(4.2) \quad E(\text{var}(\hat{\mathbf{h}}_{\mathbf{c},\text{en}})) \approx \frac{1}{C} \sum_c \left(\frac{1}{R-1} \sum_r \left(\hat{\mathbf{h}}_{\mathbf{c},r,\text{en}} - \frac{1}{R} \sum_r \hat{\mathbf{h}}_{\mathbf{c},r,\text{en}} \right)^2 \right)$$

donde C es el número de espacios de tiempo c de una realización de 5000 bits, R el número total de realizaciones, $\hat{\mathbf{h}}_{\mathbf{c},\text{en}}$ es el estimador $\hat{\mathbf{h}}_{\mathbf{c}}$ para un ruido tal que $E_b/N_0 = en$ y la función $E(\cdot)$ representa la esperanza. Debido a que $\hat{\mathbf{h}}_{\mathbf{c}} = [\hat{h}_{0,c} \hat{h}_{1,c} \dots \hat{h}_{L,c}]^T$ es un vector de L elementos la esperanza y varianza también serán vectores:

$$(4.3) \quad E[\hat{\mathbf{h}}_{\mathbf{c},\text{en}}] = \begin{bmatrix} E[\hat{h}_{0,c,\text{en}}] \\ E[\hat{h}_{1,c,\text{en}}] \\ \vdots \\ E[\hat{h}_{L,c,\text{en}}] \end{bmatrix}, \quad \text{var}(\hat{\mathbf{h}}_{\mathbf{c},\text{en}}) = \begin{bmatrix} \text{var}[\hat{h}_{0,c,\text{en}}] \\ \text{var}[\hat{h}_{1,c,\text{en}}] \\ \vdots \\ \text{var}[\hat{h}_{L,c,\text{en}}] \end{bmatrix}$$

El canal utilizado en las simulaciones es representado casi totalmente por solo una componente $h_\ell[n]$, es decir $L = 0$. Sin embargo se utilizó $L = 2$ para el estimador por lo que los parámetros

estimados $h_\ell[n]$ para $\ell > 0$ debieran ser cercanos a 0. En la figura 4.1 se muestra el valor medio del estimador para un $E_b/N_0 = 15[\text{dB}]$ lo que representa un $\text{SNR} \approx 3[\text{dB}]$.

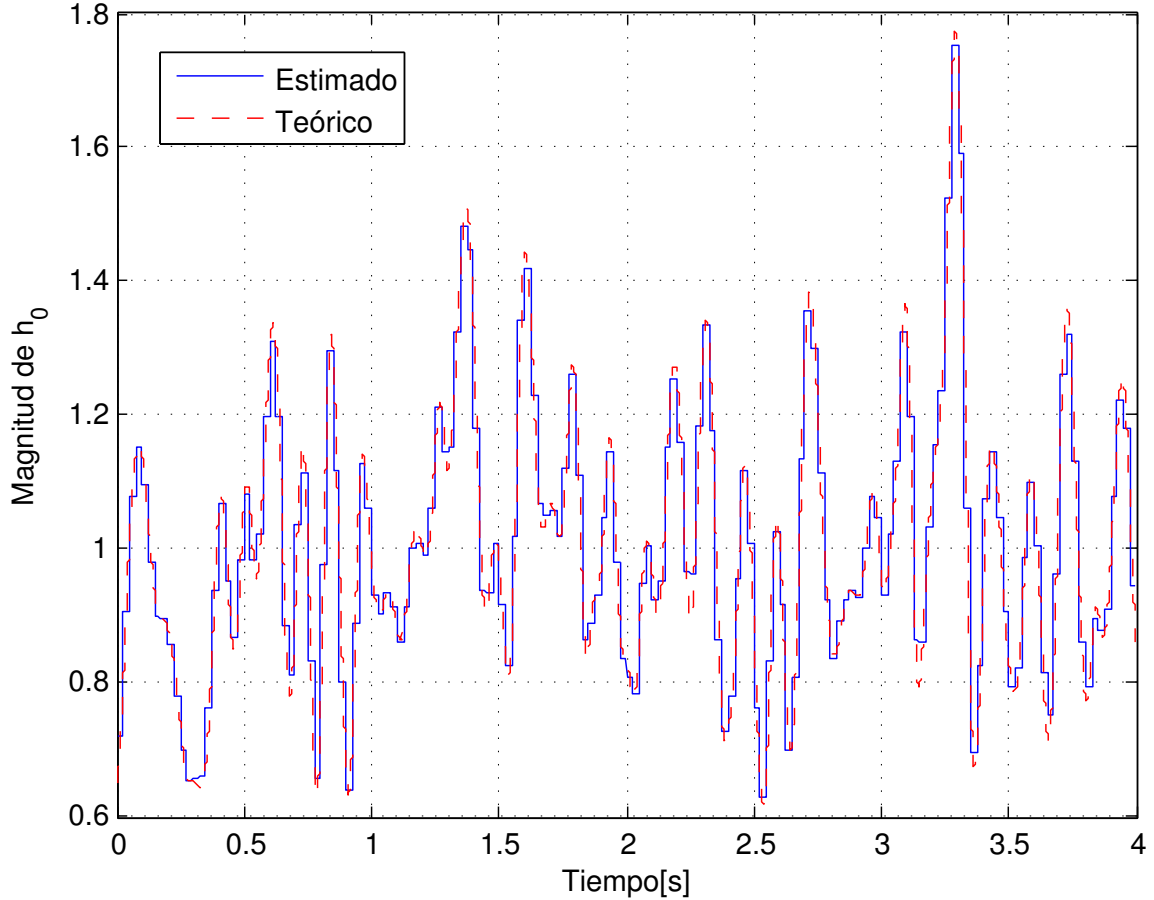


Figura 4.1: Magnitud de $h_0[n]$. La línea punteada es el valor teórico de la magnitud de $h_0[n]$ del canal Riciano fijado para la simulación y la línea continua es el valor calculado a través de la ecuación 4.1 con las estimaciones obtenidas de 100 realizaciones, utilizando el algoritmo mostrado en la sección 3.4.1 con el estimador lineal. El código de la simulación se encuentra en el archivo `algoritlin.m`

Se aprecia en la figura que el estimador corresponde a una función escalonada ya que, según como se ha definido anteriormente, este permanece constante para cada espacio de tiempo c y se actualiza para cada intervalo.

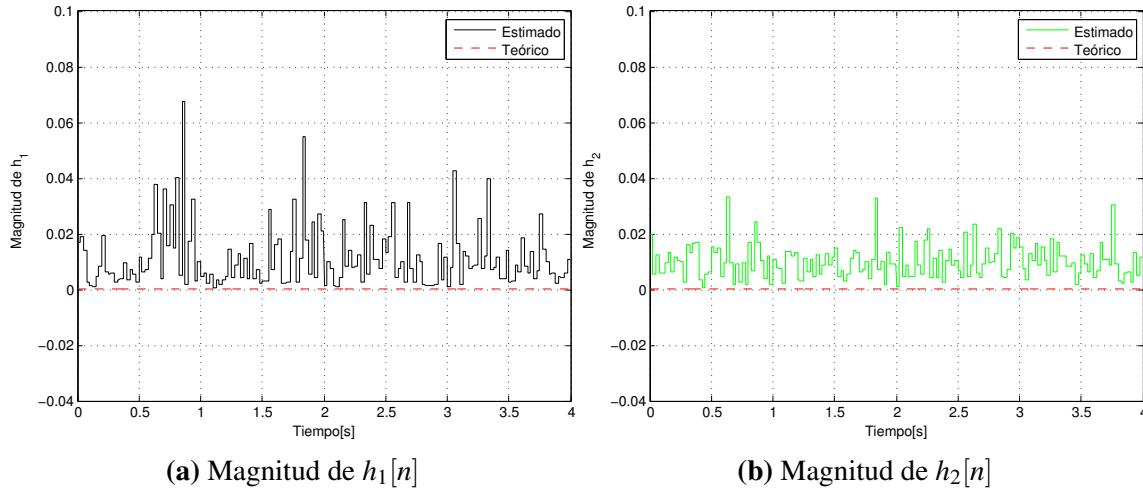


Figura 4.2: Magnitud de $h_\ell[n]$, $\ell > 0$. La metodología de obtención de datos es idéntica a la utilizada en la figura 4.1 pero se exponen los datos de $h_1[n]$ y $h_2[n]$.

Los valores medios estimados de $|h_\ell[n]|$ para $\ell > 0$ con un $E_b/N_0 = 15[db]$ se muestran en las figuras 4.2. Los valores teóricos fijados de estos valores son prácticamente 0 y se aprecia que el valor de los estimadores no supera 0.08 para $|h_1[n]|$ y 0.04 para $|h_2[n]|$ que son valores cercanos a 0 en comparación con el parámetro $|h_0[n]|$.

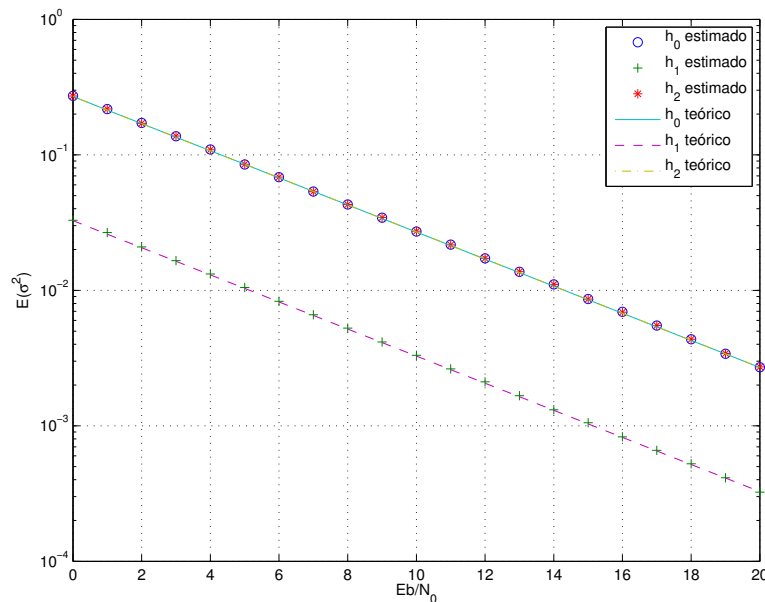


Figura 4.3: Varianza del estimador lineal. Donde los parámetros teóricos de cada estimador son calculados a través de la matriz de covarianza obtenida por la ecuación 4.4 y los valores estimados se obtienen a través de la ecuación 4.2. La simulación se realizó implementando el algoritmo de la sección 3.4.1.

$$(4.4) \quad \mathbf{C}_{\hat{\mathbf{h}}_c} = (\mathbf{X}_c^T \cdot \mathbf{X}_c)^{-1} N_0.$$

La varianza teórica y estimada del estimador se muestran en la figura 4.3. La diagonal de la matriz de covarianza representa la varianza teórica de $\hat{\mathbf{h}}_c$ para cada espacio de tiempo c . Promediando los valores para todo c se obtiene el valor teórico mostrado en la figura para cada E_b/N_0 . Este valor se compara con el obtenido por la simulación a través de la ecuación 4.2. En la práctica, el número de estimaciones que se realiza por cada E_b/N_0 asciende a $\text{realizaciones} \cdot C = 16.000$, que es un número lo bastante alto para obtener una buena aproximación de la varianza del estimador. Esto se ve representado en la figura donde se aprecia que la varianza aproximada a través de las simulaciones es prácticamente igual a la varianza teórica del estimador, que es también la cota de Cramer-Rao para estimar modelos lineales.

La varianza del estimador garantiza, que en general, habrá poca dispersión del estimador frente a su valor medio. Sin embargo, es necesario analizar si el valor medio del estimador representa bien al canal simulado. Una forma de medir esto es calculando el error cuadrático medio entre el valor medio del estimador y el valor teórico del canal simulado:

$$(4.5) \quad MSE_{en} = E \left[(E[\hat{\mathbf{h}}_{c,en}] - \mathbf{h}_c)^2 \right] \approx \frac{1}{C} \sum_c \left(\frac{1}{R} \cdot \sum_r (\hat{\mathbf{h}}_{c,r,en}) - \mathbf{h}_c \right)^2.$$

En la figura 4.4 se muestra el valor de MSE_{en} para cada parámetro h_ℓ . Los valores de la figura se pueden interpretar como que en general no habrá una diferencia mayor a 0,1 entre la media del estimador y el valor real de la variable h_ℓ para cualquier E_b/N_0 entre 0 y 20. Para las otras variables, la diferencia, en general, es aún menor a 0,1 alcanzando valores de 0,01 cuando la relación señal a ruido es alta.

La varianza obtenida para el estimador garantiza que será el óptimo (MVU) si se considera el canal como un modelo lineal y el error cuadrático medio para el valor medio del estimador garantiza que en general el estimador se parecerá al valor real. Por estas dos razones se puede considerar el estimador como una buena alternativa para canales que tengan características similares al modelo simulado.

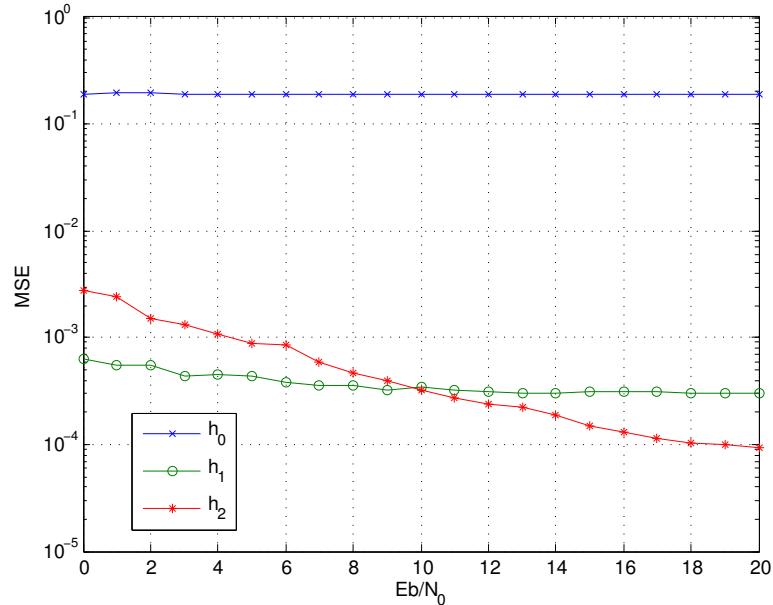


Figura 4.4: Error cuadrático medio entre los valores estimados y teóricos. Cada función representa el MSE, calculado a través de la ecuación 4.5, para cada “tap” ℓ . La simulación se realizó implementando el algoritmo presentado en la sección 3.4.1. El código de la simulación se encuentra en el archivo `algoritlin.m`

4.1.2. Comportamiento del canal estimado

Se obtuvieron características generales del canal simulado utilizando los valores del estimador lineal. En particular, se mostrará los resultados de la correlación temporal entre los estados $h_0[n]$ y los parámetros K y Ω de la distribución de Riciana.

Correlación de estados

La figura 4.5 muestra la comparación entre la correlación de los estados del canal Riciano simulado y la correlación de los estados estimados por el estimador lineal para un $E_b/N_0 = 15[db]$ (o $SNR = 3[db]$). Como es de esperarse, la correlación de los estados del canal ideal teórico empieza teniendo valor unitario y baja gradualmente hasta estabilizarse en un valor cercano a 0. Lo que quiere decir que los valores cercanos tienen un gran parecido entre ellos, que baja a medida que transcurre un tiempo suficiente, lo cual es parecido al concepto de tiempo de coherencia. En la figura 4.5 se aprecia que la correlación para los valores estimados se mantienen constantes para el tiempo de coherencia y luego cambia regularmente de acuerdo a la actualización de $\hat{\mathbf{h}}_c$ siguiendo la línea de la correlación de los estados teóricos. Se aprecia que durante el intervalo transcurrido en el tiempo de coherencia, la correlación no baja de 0,8 lo que significa que el estado no cambia sustancialmente.

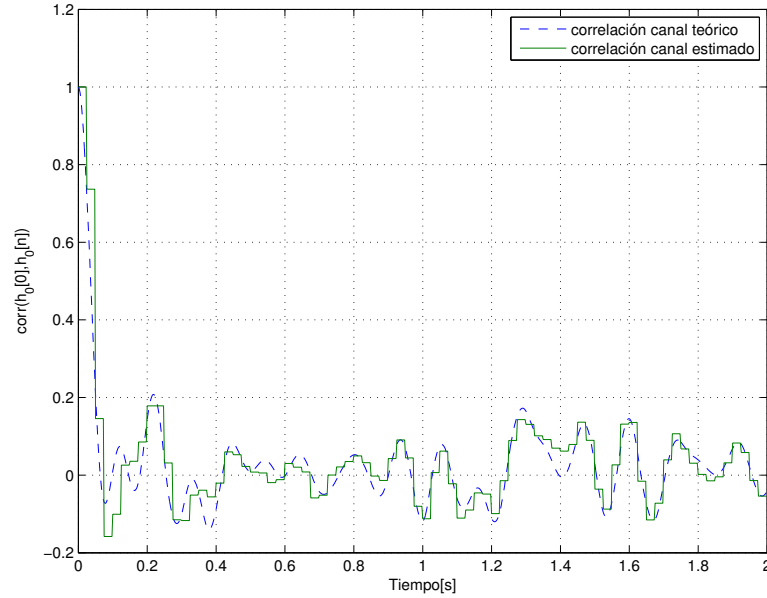


Figura 4.5: Correlación de estados $h_0[n]$. La línea punteada representa la correlación temporal del estado h_0 del canal estimado mientras la línea continua representa la correlación del mismo estado para el canal estimado. La simulación se realizó implementando el algoritmo mostrado en la sección 3.4.2. El código de la simulación se encuentra en el archivo `matrizcorrlin.m`

Parámetros K y Ω

Es posible calcular los parámetros K y Ω de la distribución Riciana a partir de los parámetros $h_\ell[n]$ estimados. Tomando en cuenta que la media y varianza de una variable aleatoria Riciana mantiene la siguiente relación:

$$(4.6) \quad \begin{aligned} \text{Media} &= \sigma \sqrt{\frac{\pi}{2}} L_{1/2} \left(\frac{-v^2}{2\sigma^2} \right) \\ \text{Var} &= 2\sigma^2 + v^2 - \frac{\pi\sigma^2}{2} L_{1/2}^2 \left(\frac{-v^2}{2\sigma^2} \right) \end{aligned}$$

La función $L_{1/2}(\cdot)$ denota un polinomio de Laguerre de orden $1/2$. Los parámetros v y σ son estadísticos de la función de distribución Riciana y es posible calcular K y Ω a través de estos estadísticos:

$$(4.7) \quad \begin{aligned} K &= \frac{v^2}{2\sigma^2} \\ \Omega &= v^2 + 2\sigma^2 \end{aligned}$$

De ésta forma es posible obtener K y Ω a través de la varianza y media del valor estimado. El cálculo de estas relaciones se hizo utilizando el paquete de estimación Riciana de Matlab que utiliza estas

ecuaciones para obtener los parámetros deseados. Los resultados obtenidos para 100 realizaciones de la simulación se muestran en las figuras 4.6a y 4.6b.

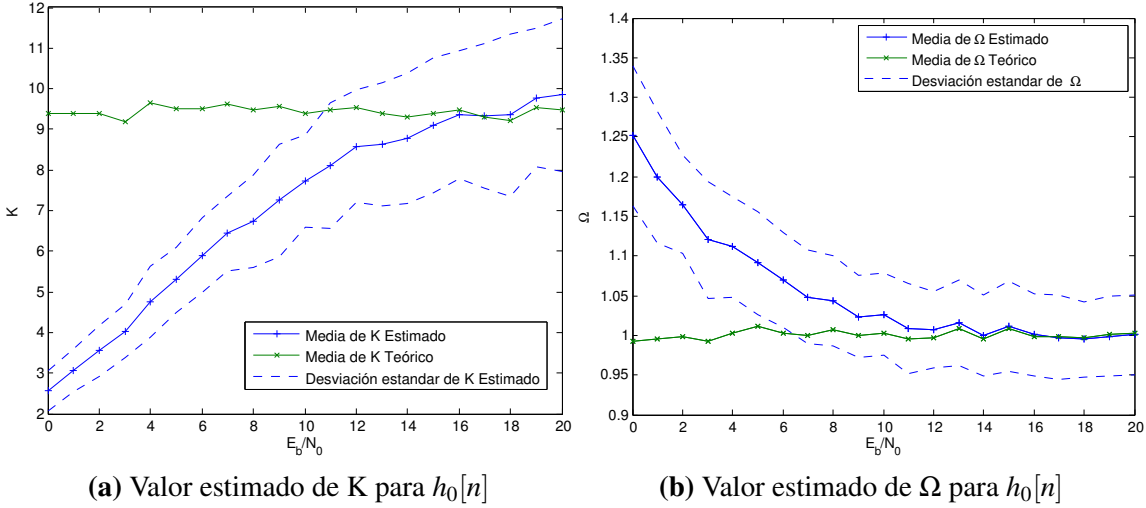


Figura 4.6: Estimación de Parámetros K y Ω . Los datos se obtuvieron realizando el algoritmo de simulación presentado en la sección 3.4.2 . Se estimó los parámetros K y Ω a través del paquete “Rician” de Matlab, para cada realización, y se calculó su promedio y desviación standar.

Los valores teóricos de K y Ω rondan entorno a 1 y 9, respectivamente, que son los parámetros que se fijaron para la simulación desde un principio. Para el parámetro K se observa que no se obtienen buenos resultados para valores de E_b/N_0 bajos, llegando a estabilizarse cuando se tienen valores de E_b/N_0 altos (alrededor de $12[dB]$). Esto se atribuye a la varianza mayor del estimador \hat{h}_ℓ que se tiene cuando E_b/N_0 es bajo. En cambio, cuando E_b/N_0 es mayor, la varianza de \hat{h}_ℓ disminuye y es posible obtener una mejor estimación de K , para Ω se observa el mismo comportamiento pero la estabilización se produce para E_b/N_0 mas bajos.

4.1.3. Secuencias piloto

Para medir el rendimiento del estimador frente al largo de la secuencia que es utilizada para estimar se comparó el valor MSE utilizando diferentes largos de secuencia. Los resultados de ésta simulación para el estado $h_0[n]$ se muestran en la figura 4.7. Se observa que empieza a haber una diferencia notable desde las 50 o menos muestras llegando a haber un MSE cercano a 1 cuando se utilizan 5 muestras, lo cual, considerando que el canal teórico oscila entre 0,4 y 1,8 para el estado $h_0[n]$ se trata de un error cercano al 100%. Tomando éstas consideraciones se observa que para obtener un error menor al 20% para el estado $h_0[n]$ del canal de la simulación, se debe escoger una secuencia piloto de al menos 100 muestras, considerando que la frecuencia de muestreo es de

$20.000[Hz]$, es necesaria una secuencia de al menos $5[ms]$ con una periodicidad de $T_c = 25[ms]$. Para una dispersión de Doppler mayor, la periodicidad de la secuencia deberá tener mayor duración.

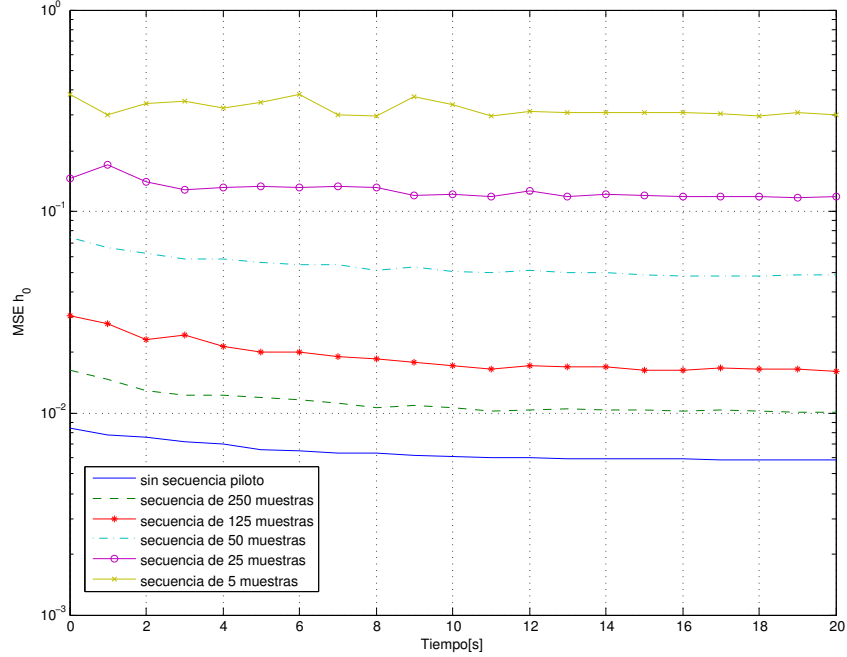


Figura 4.7: MSE de estado $h_0[n]$ para diferentes secuencias. La simulación se realizó implementando el algoritmo presentado en la sección 3.4.1 pero utilizando múltiples largos de secuencia. El valor MSE se calculó a través de la ecuación 4.5.

4.2. Resultados estimador secuencial

4.2.1. Comportamiento del estimador

Se realizó el mismo procedimiento que el utilizado para analizar el estimador lineal pero utilizando el algoritmo de estimación secuencial. La forma de obtener el valor medio del estimador y el error cuadrático medio entre la media del estimador y el valor teórico es el mismo que en el caso anterior. Sin embargo, la varianza del estimador variará en relación a n siguiendo el valor teórico:

$$(4.8) \quad \mathbf{C}_{\hat{\mathbf{h}}_c[n]} = \mathbf{C}_c[n] \cdot N_0$$

donde $\mathbf{C}_{\hat{\mathbf{h}}_c[n]}$ es la matriz de covarianza del estimador $\hat{\mathbf{h}}_c[n]$. Para obtener una comparación general entre la varianza teórica y la estimada de las 100 realizaciones, se promedió estos valores en

relación a n . Es decir:

$$(4.9) \quad \begin{aligned} E(\text{varteo}(\hat{\mathbf{h}}_{\mathbf{c},\text{en}}[n])) &= \frac{N_0(en)}{N} \sum_n^N \text{diag}(C_c[n]) \\ E(\text{varestim}(\hat{\mathbf{h}}_{\mathbf{c},\text{en}}[n])) &= \frac{1}{N} \sum_n^N \left(\frac{1}{R-1} \sum_r (\hat{\mathbf{h}}_{\mathbf{c},\text{r},\text{en}}[n] - \frac{1}{R} \sum_r \hat{\mathbf{h}}_{\mathbf{c},\text{r},\text{en}}[n])^2 \right) \end{aligned}$$

En la figura 4.8 se muestra la comparación entre el valor medio estimado para el estado $h_0[n]$ y su valor teórico para un $E_b/N_0 = 15$. Nuevamente, se considera $L = 2$ y se observa en las figuras 4.9a y 4.9b que los estados h_ℓ para $\ell > 0$ son despreciables y cercanos a cero.

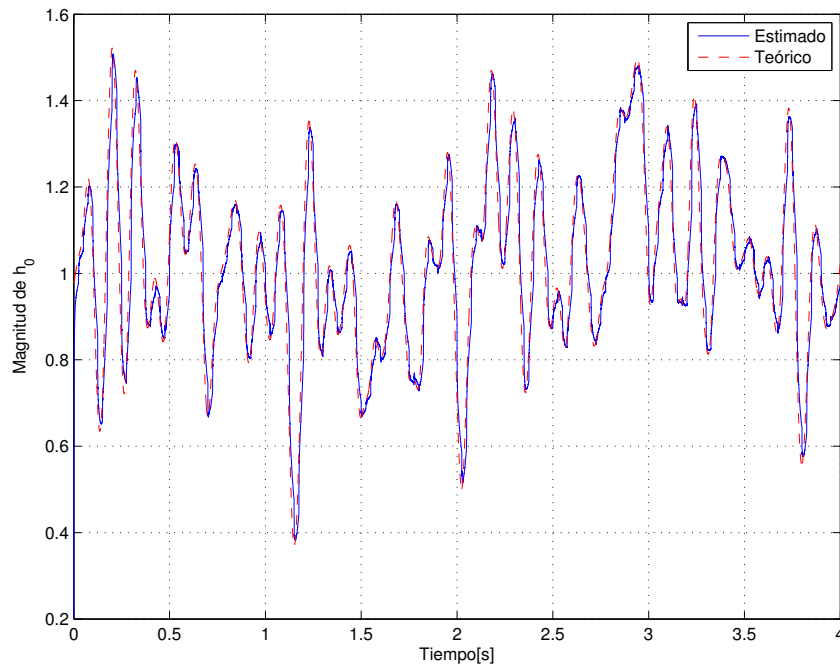


Figura 4.8: Magnitud de $h_0[n]$ para estimador secuencial para $E_b/N_0 = 15[\text{dB}]$. La simulación se desarrolló implementando el algoritmo presentado en la sección 3.4.1 utilizando el estimador secuencial. El código de la simulación se encuentra en el archivo `algoritsec.m`.

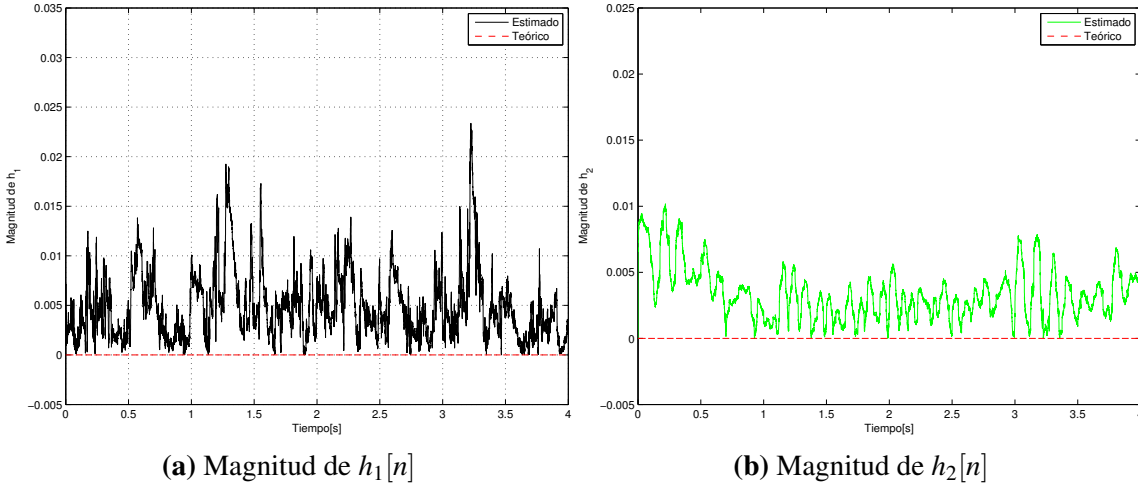


Figura 4.9: Magnitud $h_\ell[n]$, $\ell > 0$ para un $E_b/N_0 = 15[dB]$. Se utiliza la misma simulación de la figura 4.8 pero se estiman los otros estados.

Debido al procesamiento secuencial, el valor estimado ya no es una función escalón sino una función continua en el tiempo. Es importante notar que para el algoritmo secuencial el resultado obtenido en la estimación tiene relación con los valores iniciales que se escojan para $\mathbf{C}_c[cM - 1]$, ya que este valor representa la incertidumbre que se tiene sobre el valor estimado en los tiempos $n = CM - 1$. Se escogió este valor teniendo en cuenta el orden de magnitud para cada estado ℓ . En el caso que no se conozca el orden de magnitud de los estados, es posible ocupar un valor arbitrario como valor inicial pero se obtendrá un resultado peor.

El valor del **MSE** para cada estado ℓ en relación al nivel de ruido medido por E_b/N_0 se muestra en la figura 4.10. Para h_0 , el MSE es un poco mayor que para el caso del estimador lineal, pero no llega a ser significativo. Para los estados h_1 y h_2 se obtiene un MSE menor al obtenido con el estimador lineal. Esta disminución se debe a la elección del valor $\mathbf{C}_c[cM - 1]$, sin embargo, como el valor de éstas constantes es despreciable, no llega a ser un factor de decisión sobre un estimador u otro.

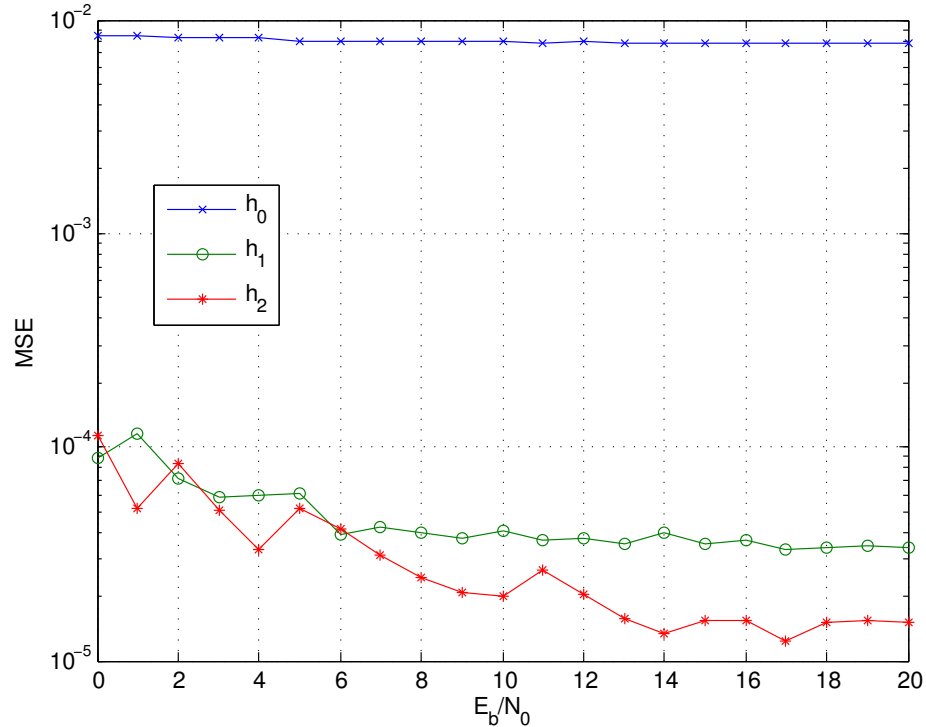


Figura 4.10: Error cuadrático medio entre los valores estimados y teóricos; Cada función representa el MSE, calculado a través de la ecuación 4.5, para cada “tap” l . La simulación se realizó implementando el algoritmo presentado en la sección 3.4.1, utilizando el estimador secuencial. El código de la simulación se encuentra en el archivo `algoritsec.m`.

La figura 4.11 muestra la comparación entre la varianza estimada de las realización de $h_\ell[n]$ y la varianza teórica obtenida a través de $\mathbf{C}_c[\mathbf{n}]$. Se observa en los resultados una menor varianza media para los valores estimados. Esto se atribuye a que la varianza teórica es calculada asumiendo que el canal es lineal, lo cual estrictamente no es cierto. Por esta razón, al hacer el cálculo secuencial, los valores teóricos difieren con los prácticos. Sin embargo, la curva de para todos los estados es similar cuando se compara el comportamiento teórico con el estimado.

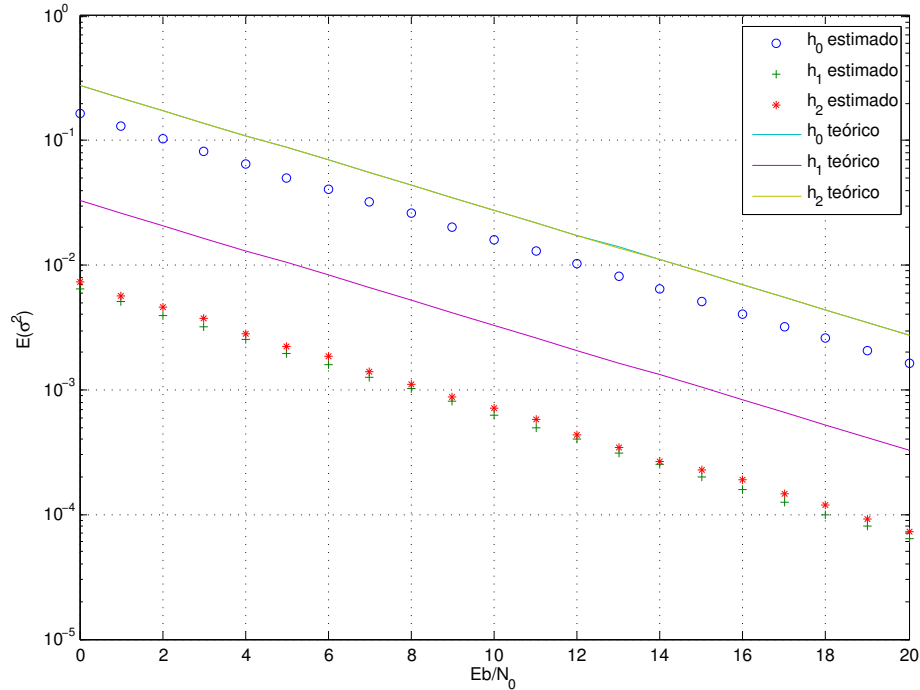


Figura 4.11: Varianza del estimador secuencial. Donde los parámetros teóricos y estimados son calculados a través de las ecuaciones 4.9. La simulación se realizó implementando el algoritmo de la sección 3.4.1.

4.2.2. Comportamiento del canal estimado

El análisis del comportamiento del canal se realizó con las mismas herramientas utilizadas para el estimador lineal, pero aplicadas al estimador secuencial.

Correlación de estados

Se obtuvo la correlación en el tiempo del estado $h_0[n]$ para el estimador secuencial. Los resultados para un $E_b/N_0 = 15[dB]$ se muestran en la figura 4.12. La correlación del canal estimado para los primeros estados muestra un valor significativamente menor a 1. Esto se explica debido a que para $n = 0$ la secuencia de estimación está empezando y se debe confiar demasiado en los valores fijados arbitrariamente en las condiciones iniciales de los estados. Sin embargo de todas formas el valor estimado tiene un forma parecida al valor teórico, con un valor mayor para el tiempo $n = 0$ y estabilizándose en torno a 0 a medida que transcurre el tiempo.

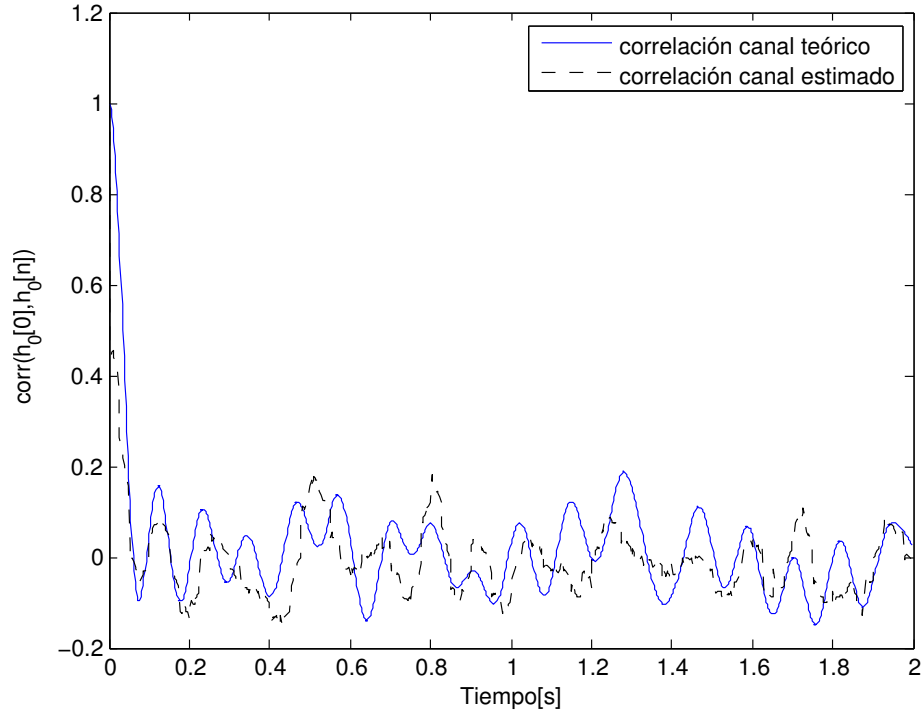


Figura 4.12: Correlación de estados $h_0[n]$ utilizando el estimador secuencial. La línea punteada representa la correlación temporal del estado h_0 del canal estimado mientras la línea continua representa la correlación del mismo estado para el canal estimado. La simulación se realizó implementando el algoritmo de la mostrado en la sección 3.4.2. El código de la simulación se encuentra en el archivo `matrizcorrsec.m`

Estimación de K y Ω

Los resultados para la estimación de los parámetros K y Ω utilizando el algoritmo secuencial se presentan en las figuras 4.13a y 4.13b. En comparación al esquema lineal la estabilización para ambos parámetros, frente al valor real, se produce para valores menores de E_b/N_0 . Esto se explica debido a que la varianza para el estimador secuencial es menor a la del lineal, lo cual desemboca en una mejor estimación del estado lo que se traduce en una mejor evaluación de los parámetros K y Ω para E_b/N_0 bajos.

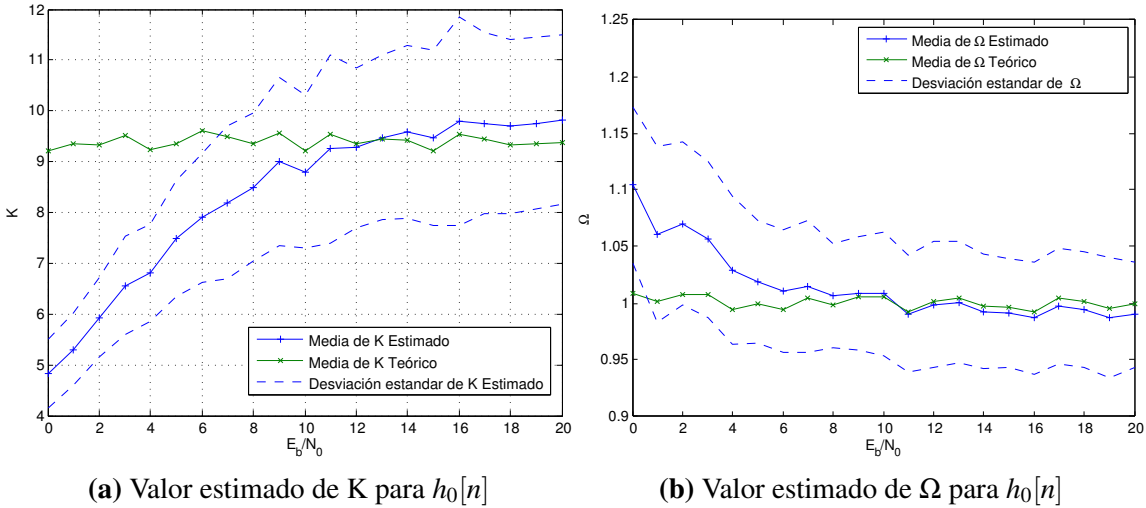


Figura 4.13: Estimación de Parámetros K y Ω utilizando el estimador secuencial. Los datos se obtuvieron realizando el algoritmo de simulación presentado en la sección 3.4.2. Se estimó los parámetros K y Ω a través del paquete “Rician” de Matlab, para cada realización, y se calculó su promedio y desviación standar.

4.2.3. Secuencias Piloto

Se realizó un análisis del comportamiento del estimador secuencial frente al largo de la secuencia piloto usada para estimar. Los resultados para el estado más significativo ($h_0[n]$) se muestran en la figura 4.14. Para la secuencia menor, de 5 muestras se obtuvo un MSE inferior a 0.08 lo que significa un error medio menor al 28% y para obtener un error menor al 20% se deben elegir secuencias de 50 muestras o más. En comparación al estimador lineal son necesarias menos muestras para obtener valores iguales de MSE, esto se atribuye a que el algoritmo secuencial no invierte matrices a diferencia del estimador lineal que si lo hace y puede conllevar a errores para matrices cercanas a singulares

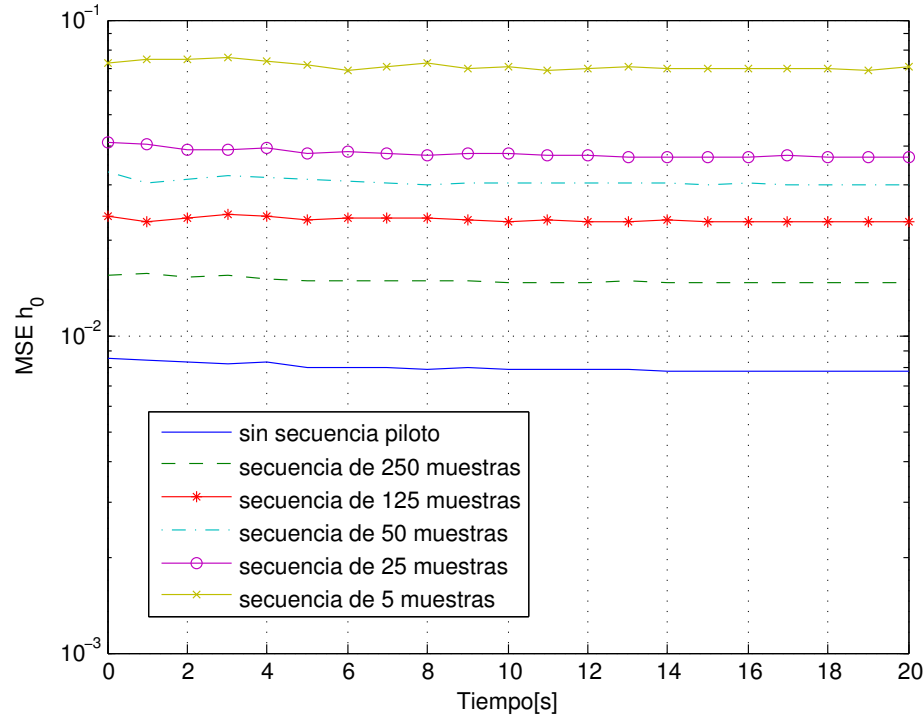


Figura 4.14: MSE de estado $h_0[n]$ para diferentes secuencias utilizando el estimador secuencial. La simulación se realizó implementando el algoritmo presentado en la sección 3.4.1 pero utilizando múltiples largos de secuencia. El valor MSE se calculó a través de la ecuación 4.5.

4.3. Comparación de estimadores

Los valores de MSE para el estado significativo $h_0[n]$ para ambos estimadores son similares cuando se utiliza todo el mensaje para estimar. Sin embargo el análisis de secuencias pilotos demuestra que el estimador secuencial tiene mejor desempeño para secuencias iguales y es posible obtener resultados aceptables (cerca de un 20% de error) para secuencias pequeñas de hasta 25 muestras.

La varianza de ambos estimadores es parecida al valor teórico esperado para cada uno, siendo la varianza del estimador secuencial menor para todos los estados $h_\ell[n]$. Esto se explica debido a la elección adecuada de condiciones iniciales (cerca del rango de magnitud de cada estado).

La estimación de parámetros K y Ω en general es más acertada para el estimador secuencial, principalmente, por la varianza menor que posee.

En todos los campos el estimador secuencial parece ser una mejor opción que el estimador lineal. La razón de esto, puede explicarse, en que el estimador lineal necesita invertir la matriz

de observaciones, a diferencia del caso secuencial. Esto lo hace menos susceptible a fallas en la estimación cuando la matriz de observaciones tiene un rango de fila (M) pequeño y se vuelve casi singular.

4.4. Implementación a canal real

Se desarrollaron funciones en matlab implementando los estimadores lineal y secuencial (hlineal.m y hsec.m) a los datos de una señal cualquiera. El funcionamiento de ambas funciones se explica en el esquema 4.15.

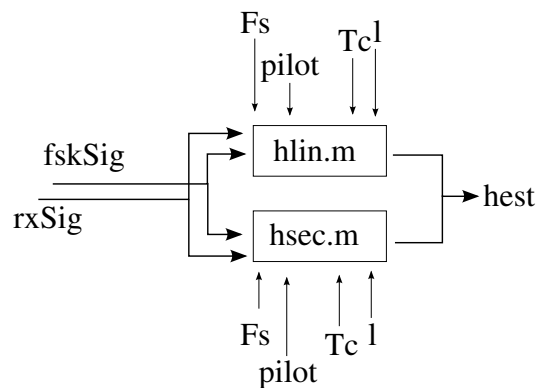


Figura 4.15: Esquema de entradas y salida de funciones hlineal.m y hsec.m

Ambas funciones entregan la caracterización del canal, usando los estimadores lineal o secuencial respectivamente, para cada instante de tiempo muestreado. Las entradas necesarias pueden ser obtenidas directamente de los dispositivos del esquema de comunicaciones, o se pueden aproximar de otras variables. La obtención de cada parámetro se explicará a continuación:

- **Señal enviada:**

Es posible obtener la señal enviada a través del mensaje digital que tienen las secuencias piloto. Debido a que se sabe que la modulación es FSK binaria, basta conocer la frecuencia de separación y de muestreo del modulador para obtener la señal análoga emitida desde la antena, luego, esta señal debe ser muestreada y normalizada a potencia unitaria de forma de obtener la señal de entrada **fskSig**.

Mientras los parámetros del modulador (Frecuencia de separación y muestreo) se mantengan constantes los valores de la entrada **fskSig** puede considerarse constante y mantener sus valores en un registro.

- **Señal recibida:**

Esta señal análoga se debe obtener directamente desde la antena satélital o terrestre,

dependiendo de si el enlace es de subida o bajada respectivamente. La señal debe muestrearse a la misma frecuencia definida por el modulador normalizar su potencia a 1. Estos valores definen la señal **rxSig**, necesaria para realizar la estimación del canal.

- **Frecuencia de muestreo:**

La frecuencia de muestreo **F_s** se obtiene desde el modulador y es un valor constante de la estimación.

- **Tiempo de coherencia:**

Es un parámetro que no se conoce a priori, pero se puede aproximar utilizando la matriz de covarianza mostrada en la sección 4.1.2. En una primera aproximación se realiza la estimación del canal (lineal o secuencial) múltiples veces (100 experimentos) fijando un tiempo de coherencia pequeño y calculando la matriz de covarianza con estos datos. Como se observa en la figura 4.5, el tiempo de coherencia aproximado del canal será la diferencia de tiempo para el cual la autocovarianza temporal baje desde 1 hasta 0,7. Como se observó anteriormente, analizando las secuencias piloto, para obtener una buena estimación de los parámetros del canal son necesarias alrededor de 100 muestras, lo cual representa 5[ms] (alrededor de 50[Hz] de dispersión de Doppler) con la frecuencia de muestreo de la simulación. Este valor es el menor tiempo de coherencia para el cual se obtendrá una estimación adecuada a una frecuencia de muestreo de 20[KHz]. Si luego de realizar la primera caracterización se observa que el canal tiene un tiempo de coherencia menor a 5[ms] sería necesario aumentar la frecuencia de muestreo del modulador para obtener más muestras con las cuales estimar, sin embargo se estima que la dispersión de Doppler no debiera ser alta (sección 2.3.6).

- **Rango de la matriz de observación**

Se debe definir el rango **l** de la matriz de observaciones. Para una primera aproximación se debe elegir un rango elevado (alrededor de 10) y luego disminuir este valor a medida que se observe que los parámetros de mayor rango sean cercanos a cero.

- **Largo de la secuencia piloto:**

Una vez definido el tiempo de coherencia es posible realizar experimentos (de la misma forma que la sección 3.4.1) para obtener el mínimo largo necesario. Para las primeras estimaciones es posible designar el largo de la secuencia piloto igual al largo del tiempo de coherencia y enviar mensajes solo con el fin de estimar el canal y aproximar los diferentes parámetros necesarios (largo de secuencia piloto, tiempo de coherencia, rango de matriz de observaciones).

La forma de implementar las funciones `hlineal.m` y `hsec.m` en el esquema satelital, depende de si el enlace es de bajada o de subida

Enlace de bajada

En el enlace de bajada las funciones de estimación se deben procesar en el servidor terrestre y la señal **rxSig** se obtiene de la antena terrestre. Debido a que las funciones se correrán en un computador servidor se podrán realizar directamente desde Matlab. Tal como se muestra en el esquema 4.16. La señal muestreada debe guardarse en un arreglo y pasarse como argumento a la función de estimación que se utilizará. Los demás parámetros de las funciones pueden considerarse constantes.

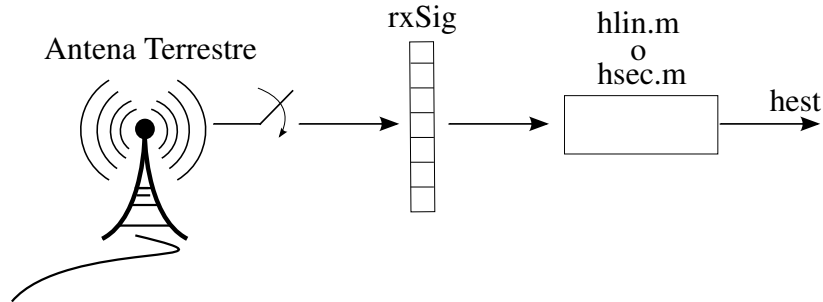


Figura 4.16: Forma de obtener la entrada **rxSig** para enlace de bajada

Enlace de subida

Para el enlace de subida las funciones deberán procesarse desde el computador a bordo del cubesat (microcontrolador PIC), por lo que se necesitaría adaptar las funciones al lenguaje de programación puntual que se utilice en el PIC. Otra opción sería a través de simulink implementar las funciones en una FPGA y adjuntarla a los experimentos a bordo, sin embargo, esta alternativa tendría que evaluarse para misiones futuras junto con esquemas de adaptación de señal al canal.

Métricas de desempeño en el canal real

Una vez implementadas las funciones que contienen los estimadores, es necesario desarrollar una métrica para definir si la estimación es adecuada o no. Debido a que los parámetros h_ℓ reales no son conocidos, no es posible comparar los valores estimado con un valor teórico (como se hizo anteriormente), sin embargo es posible realizar la comparación, por ejemplo a través del MSE, de la señal recibida estimada con respecto a la recibida realmente.

$$MSE(y[n]) = (y[n] - \hat{y}[n])^2 = (y[n] - \sum_{\ell=0}^L \hat{h}_\ell[n]x[n-\ell])^2$$

$$\Rightarrow E(MSE(y[n])) \approx \frac{1}{N} \sum_n (y[n] - \sum_{\ell=0}^L \hat{h}_\ell[n]x[n-\ell])^2$$

Capítulo 5

Conclusiones

El desempeño mostrado por los estimadores implementados demuestra que son una alternativa válida para caracterizar un canal de comunicaciones con los parámetros de la simulación utilizada. Las ventajas comparativas del estimador secuencial sobre el lineal, como la facilidad de implementación y el mejor desempeño frente a diferentes secuencias piloto, hacen de este estimador la mejor opción para implementar en un esquema real.

La investigación teórica permitió obtener madurez sobre los conocimientos teóricos necesarios para realizar una simulación y caracterización del canal representativa de la realidad. Los conocimientos obtenidos sobre estadística y procesamiento estadístico de señales permitieron definir métricas de calidad y comparación entre los esquemas realizados.

La obtención de un modelo matemático representativo de un canal inalámbrico permitió, realizando suposiciones en relación al modelo, realizar algoritmos de estimación capaces de caracterizar el canal supuesto. La realización e implementación de una simulación representativa del esquema de comunicaciones del proyecto Suchai permitió un mecanismo ideal para probar el desempeño de los estimadores propuestos.

El análisis realizado con las secuencias piloto demuestra, que para caracterizar un canal con las propiedades de la simulación se necesitan a lo menos 100 muestras para el caso del estimador lineal y 50 para el caso secuencial, con un error medio menor al 20%, lo que significa alrededor de 5[ms] y 2,5[ms] para una frecuencia de muestreo de 20[KHz]. Esto respresenta, que el tiempo de coherencia mínimo para el cual los estimadores realizarán una buena caracterización, es de estas magnitudes de tiempo.

La implementación de los algoritmos de simulación y estimación permitieron obtener el desempeño de cada estimador frente a diferentes parámetros de medición. El desempeño de ambos estimadores se comparó cualitativamente y se desarrolló una conclusión técnica frente a cual de las alternativas es la mejor opción a desarrollar realmente.

Se desarrollaron funciones, de fácil utilización, capaces de realizar una estimación del canal

teniendo las señales de entrada, salida y algunos parámetros característicos del canal. Se explicó el significado de cada parámetro de las funciones y la forma de obtener cada uno en un esquema real de comunicaciones. Existen ciertas características del canal, como el tiempo de coherencia, el rango de la matriz de observación y el largo de la secuencia piloto, que deben fijarse en un valor determinado para una primera caracterización del canal, pero luego se pueden adaptar utilizando un concepto de prueba y error (como se explicó en la sección 4.4).

El desarrollo de funciones capaces de estimar el canal para un par de entrada y salida determinado, posibilita su utilización directa en el esquema de enlace de bajada, donde la memoria y velocidad de procesamiento no se encuentran coartadas. Sin embargo la implementación en el enlace de subida necesita más trabajo, ya sea cambiando el código fuente de las funciones al lenguaje nativo del procesador a bordo o implementando las funciones de Matlab en una FPGA. De todas formas es posible estimar el enlace de bajada sin ninguna intervención directa al hardware a bordo del satélite, por lo que para una primera aproximación al canal real, es preferible obtener estos datos y luego evaluar esquemas de enlace de subida y “feedback” entre emisor y receptor.

En general, se ha cumplido con los objetivos propuestos para el trabajo de memoria y se han desarrollado algoritmos para obtener los estados de un canal real con las características presentes en el esquema del proyecto SUCHAI (modulación, tasa de datos, frecuencia de muestreo). A pesar que no se ha encontrado implementaciones anteriores de estos estimadores para las características del proyecto SUCHAI, los resultados obtenidos demuestran que la estimación es posible dentro de ciertos rangos (mayor a $2,5[ms]$ para tiempo de coherencia y mayor a $-4[dB]$ de relación señal a ruido). Misiones anteriores de otros satélites, normales y miniaturizados, datan que la dispersión de Doppler y la relación señal a ruido se mantienen en estos rangos (o aún más optimistas). Por lo tanto la estimación de un canal real utilizando las funciones desarrolladas es realizable.

5.1. Trabajos Futuros

Del trabajo realizado se desglosan varias alternativas de trabajo futuro:

- Investigación y desarrollo de algoritmos de estimación basados en otros supuestos realizados frente al canal. Por ejemplo existe la alternativa de asumir que el cambio del canal en el tiempo se realiza a través de una cadena de Markov Gaussiana, en este caso es posible implementar un estimador basado en filtro de Kalman [4] para caracterizar el canal. También está la alternativa de suponer que los parámetros a estimar en el canal se tratan de variables aleatorias con cierta función de distribución de parámetros conocidos, para este acercamiento es posible utilizar un estimador Bayesiano como el estimador MMSE [32] (“Minimum Mean Square Error”).
- Implementación de las funciones desarrolladas frente datos reales obtenidos con la estación terrena del Departamento de Ciencias Físicas y Matemáticas de la Universidad de Chile. Se

estima que el primer satélite realizado por la Universidad de Chile se pondrá en órbita a finales del año 2012, por lo que se puede utilizar este medio para obtener datos de primera mano.

- Investigación y desarrollo de técnicas de adaptación de señales frente al conocimiento del canal de transmisiones. Las diferentes alternativas de adaptación, tanto en potencia como en codificación, utilizando la información estimada del canal plantean una alternativa interesante de investigación.
- Desarrollo de algoritmos y procesos de estimación en línea y “feedback” entre el transmisor y receptor. Las posibles ventajas que tenga un esquema como este para satélites miniaturizados pueden estudiarse y evaluarse.

Glosario

A

AWG Additive White Gaussian Noise -.

B

BER Bit Error Rate - Tasa de error.

BFSK Binary Shift Keying -.

C

CRLB Cramer Rao Lower Bound -.

F

FIR Finite Impulse Response.

FSK Frequency Shift Keying - Sistema de modulación digital en frecuencia.

G

GEO Geostationary Earth Orbit.

GPS Global Positioning System -.

H

HEO Highly Elliptical Orbit - Satélites de baja órbita.

I

ITU International Communication Union.

L

LEO Lower Earth Orbit - Satélites de baja órbita.

LOS Line of Sight - Línea de visión.

LTI Lineal Time Invariant - Lineal e invariante en el tiempo.

M

MEO Medium Earth Orbit.

MLE Maximum Likelihood estimation -.

MSE Mean Square Error -.

MVU Minimum Variance Unbiased -.

N

NLOS Non Line of Sight - Sin línea de visión.

S

SNR Signal to Noise Ratio - Relación Señal a Ruido.

SUCHAI Satellite of University of Chile for Aerospace Investigation -.

T

TDL Taped Delay Line.

W

WSS Wide-Sense Stationary -.

Bibliografía

- [1] Alminde, L., M. Bisgaard, D. Vinther, T. Viscor y K. Ostergaard: *Robustness of Radio Link Between AAU-Cubesat and Ground Station*, 2002.
- [2] Bjornson, E. y B. Ottersten: *A Framework for Training-Based Estimation in Arbitrarily Correlated Rician MIMO Channels With Rician Disturbance*. *Signal Processing, IEEE Transactions on*, 58(3):1807 –1820, march 2010, ISSN 1053-587X.
- [3] Chen, P. y H. Kobayashi: *Maximum likelihood channel estimation and signal detection for OFDM systems*. En *Communications, 2002. ICC 2002. IEEE International Conference on*, volumen 3, páginas 1640–1645. IEEE, 2002.
- [4] Davis, L.M., I.B. Collings y R.J. Evans: *Estimation of LEO satellite channels*. En *Information, Communications and Signal Processing, 1997. ICICS., Proceedings of 1997 International Conference on*, volumen 1, páginas 15 –19 vol.1, sep 1997.
- [5] Dogandzic, A. y Jinghua Jin: *Estimating statistical properties of MIMO fading channels*. *Signal Processing, IEEE Transactions on*, 53(8):3065 – 3080, aug. 2005, ISSN 1053-587X.
- [6] Frigyes, B. Molnár: *Investigations on the satellite-to-indoor channel characteristics*, 1998. <http://docs4.mht.bme.hu/documents/jerusat2/jerusat2.html>.
- [7] Frigyes, I., Molnár B. G. Vallet R. Herczku Z. y Z. Bodnár: *Doppler spread characteristics of satellite personal communication channels*. En *ICPWC International Conference on Wireless Personal Communications*, volumen 19, páginas 251–262, june 1995.
- [8] Hant, J., V. Lin y P. Anderson: *Verification of satellite-channel simulators with scintillation, terrestrial multipath, and shadowing effects*. En *Aerospace Conference, 2005 IEEE*, páginas 1750 –1763, march 2005.
- [9] Hermansson, M. y V. Skoda: *Evaluating channel estimation methods for 802.11 p systems*. 2011.
- [10] Ippolito, L.J.: *Satellite communications systems engineering: atmospheric effects, satellite link design and system performance*. Wiley series on wireless communications and mobile

- computing. Wiley, 2008, ISBN 9780470725276. <http://books.google.cl/books?id=GSgfAQAAIAAJ>.
- [11] Iskander, C.D. y H.T. Multisystems: *A matlab-based object-oriented approach to multipath fading channel simulation*. MATLAB White Paper, 2008.
- [12] Jeong, Haeseong, Do Hoon Kim y Heung Gyoon Ryu: *Doppler spread and its compensation by FDE and turbo code*. Information Networking, International Conference on, 0:280–284, 2012.
- [13] Jeruchim, M.C., P. Balaban y K.S. Shanmugan: *Simulation of Communication Systems: Modeling, Methodology, and Techniques*. Information technology: transmission, processing, and storage. Kluwer Academic/Plenum Publishers, 2000, ISBN 9780306462672.
- [14] Jiang, D. y L. Delgrossi: *IEEE 802.11 p: Towards an international standard for wireless access in vehicular environments*. En *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, páginas 2036–2040. Ieee, 2008.
- [15] Kandangath, A.: *Simulation of frequency-flat fading channels*. IEEE-558 Wireless Communications Project, 1, 2003.
- [16] Kay, Steven M.: *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993, ISBN 0-13-345711-7.
- [17] Kim, J.G., T. Joon y J.T. Lim: *Channel estimation for OFDM over Fast Rayleigh Fading Channels*. Proceedings of world Academy of science and technology, 21:455–458, 2007.
- [18] Klofas, B., J. Anderson y K. Leveque: *A survey of cubesat communication systems*. En *5th Annual CubeSat Developers' Workshop*, 2008.
- [19] Love, D.J., R.W. Heath, V.K.N. Lau, D. Gesbert, B.D. Rao y M. Andrews: *An overview of limited feedback in wireless communication systems*. Selected Areas in Communications, IEEE Journal on, 26(8):1341 –1365, october 2008, ISSN 0733-8716.
- [20] Maral, G., M. Bousquet y Z. Sun: *Satellite Communications Systems: Systems, Techniques and Technology*. John Wiley & Sons, 2010, ISBN 9780470714584. <http://books.google.cl/books?id=FFHEEIYbecEC>.
- [21] Milojevic, M., M. Haardt, E. Eberlein y A. Heuberger: *Channel Modeling for Multiple Satellite Broadcasting Systems*. Broadcasting, IEEE Transactions on, 55(4):705 –718, dec. 2009, ISSN 0018-9316.
- [22] Morin, David: *2D waves and other topics*, 2010. <http://www.people.fas.harvard.edu/~djmorin/waves/miscellaneous.pdf>.
- [23] Morin, David: *Interference and diffraction*, 2010. <http://www.people.fas.harvard.edu/~djmorin/waves/interference.pdf>.

- [24] Proakis, John G. y Dimitris K. Manolakis: *Digital Signal Processing (4th Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006, ISBN 0131873741.
- [25] Seun, Ajayi Taiwo: *Mobile Satellite Communications*. Tesis de Doctorado, Blekinge Institute of Technology, May 2007.
- [26] Sheng, Ding, Li Limin y Liu Huijie: *Multipath Channel Measurements and Analysis at UHF Band for LEO Satellites*. En *Communications and Mobile Computing (CMC), 2010 International Conference on*, volumen 2, páginas 112 –116, april 2010.
- [27] Sreng, S., B. Escrig y M.L. Boucheret: *Adapting DVB-SH system parameters to mobile environments*. En *SPACOMM 2011, The Third International Conference on Advances in Satellite and Space Communications*, páginas 7–11, 2011.
- [28] Tse, David y Pramod Viswanath: *Fundamentals of wireless communication*. Cambridge University Press, New York, NY, USA, 2005, ISBN 0-5218-4527-0.
- [29] University, California Polytechnic State: *Cubesat Design Specification Rev. 12*, 2009. http://www.cubesat.org/images/developers/cds_rev12.pdf.
- [30] Warty, C.: *Cooperative communication for multiple satellite network*. En *Aerospace Conference, 2010 IEEE*, páginas 1 –7, march 2010.
- [31] Wolcott, T.J. y W.P. Osborne: *Uplink-noise limited satellite channels*. En *Military Communications Conference, 1995. MILCOM '95, Conference Record, IEEE*, volumen 2, páginas 717 –721 vol.2, nov 1995.
- [32] Xiao, Pei, E. Strom y R. Carrasco: *Estimation of time-varying multipath Rayleigh fading channels in asynchronous DS-CDMA systems*. En *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, volumen 4, páginas 2570 – 2575 Vol.4, sept. 2004.
- [33] Yang, Cheng Ying y Kuo Hsiung Tseng: *Error rate prediction of the low Earth orbit (LEO) satellite channel*. En *Communications, 2000. ICC 2000. 2000 IEEE International Conference on*, volumen 1, páginas 465 –469 vol.1, 2000.

Apéndice A

Código fuente

A.1. Funciones de estimación

Código Fuente: hlineal.m

```
1 function [ hest ] = hlineal(fskSig, rxSig,Fs, Tc,l,pilot )
2 %estima los parametros del canal, asumiendo que tiene tiempo de coherencia
3 %Tc. El modulador tiene frecuencia de muestreo Fs y se ocupara un numero pilot de muestras
4 %como secuencia piloto
5 %resive la entrada y salida analoga muestreada y entrega los parametros
6 %h estimados con un rango l para cada instante en que fue tomada la muestra.
7
8
9 mptc = fix(Fs*Tc); %calcula muestras que existe por el tiempo de coherencia que se tiene
10 R= fix(length(rxSig)/mptc);%R las veces que cabe el Tc en todo el tiempo de la senial
11
12 H = creaH2(fskSig,l); %calcular la matriz de observaciones H
13 [ hest,covh ] = estimador( H,rxSig,mptc,R,pilot,l); %utilizar el estimador lineal para obtener ←
    los parametros h
14
15 end
16
17 function [ thetat,covh ] = estimador( H,y,mptc,R,pilot,realizacion)
18 %estimador lineal
19 %obtiene una estimacion , para cada muestra de la senial rxSig , de los parametros h utilizando el←
    estimador
20 %lineal
21 M = size(H,2);
22 N = size(H,1);
23 thetam =zeros(M,l,R);
24 h = zeros(pilot,M,R);
25
26 ym=zeros(pilot,l,R);
27
28 covh = zeros(M,M,R+1);
29
30
31 for r= 1:R
32     h(:,:,r) = H((r-1)*mptc+1:(r-1)*mptc+pilot,:);
```



```

33     ym(:, :, r) = y((r-1)*mptc+1:(r-1)*mptc+pilot, 1);
34     aux = pinv(h(:, :, r));
35     thetam(:, :, r) = aux*ym(:, :, r); %estima
36     if (abs(thetam(1,1,r)) > 10)
37         if (r == 1)
38             thetam(:, :, r) = zeros(size(thetam(:, :, r), 1), size(thetam(:, :, r), 2));
39         else
40             thetam(:, :, r) = thetam(:, :, r-1);
41         end
42     end
43     thetat((r-1)*mptc+1:mptc*r, :) = repmat(conj(thetam(:, :, r)'), mptc, 1);
44     if (realizacion == 100)
45         covh(:, :, r) = h(:, :, r)'*h(:, :, r);
46     end
47
48
49 end
50 if (N-(R)*mptc+1 >= pilot)
51     k = (R)*mptc+pilot;
52 else
53     k = N;
54 end
55 hfin = H((R)*mptc+1:k, :);
56 yf = y((R)*mptc+1:k, 1);
57 aux = pinv(hfin);
58 thetaf = aux*yf;
59 thetat((R)*mptc+1:k, :) = repmat(conj(thetaf(:, :)'), N-R*mptc, 1);
60 if (realizacion == 1)
61     covh(:, :, R+1) = hfin'*hfin;
62 end
63 end
64
65 function [H] = creaH2(x, l)
66 %transforma la senial de entrada en la matriz de observaciones H
67 %x es la senial analogica muestreada y l el rango de parametros h.
68 N = length(x);
69
70 H = zeros(N, l);
71 for i = 1:N
72     % disp(i)
73     if (i > l-1)
74         H(i, :) = fliplr(conj(x(i-l+1:i, :)))';
75     else
76         H(i, 1:i) = fliplr(conj(x(1:i)))';
77     end
78 end
79
80 end

```

Código Fuente: hsec.m

```

1 function [hest] = hsec(fskSig, rxSig, Fs, Tc, l, pilot)
2 %estima secuencialmente los parametros del canal, asumiendo que tiene tiempo de coherencia
3 %Tc. El modulador tiene frecuencia de muestreo Fs y se ocupara un numero pilot de muestras
4 %como secuencia piloto
5 %resive la entrada y salida analogica muestreada y entrega los parametros
6 %h estimados con un rango l para cada instante en que fue tomada la muestra
7
8
9
10 mptc = fix(Fs*Tc); %calcula muestras que existe por el tiempo de coherencia que se tiene
11 R = fix(length(rxSig)/mptc); %R las veces que cabe al Tc en todo el tiempo de la senial

```

```

12
13 H = creaH2(fskSig,1); %calcular la matriz de observaciones H
14 [ hest,Sig] = estimadorseq( H,rxSig,mptc,R,1,pilot);
15 end
16
17 function [ thetat,Sig] = estimadorseq( H,y,mptc,R,no,pilot)
18 %estima secuencialmente los parametros h
19
20 M = size(H,2);
21 N = size(H,1);
22
23 thetat = zeros(N,M);
24 Sigma= zeros(M,M,N);
25 h0 = ones(M,1)*0;
26 Sigma0 = diag([0.1 0.001 0.001]);
27 Sig = zeros(N,M);
28 for r= 1:R
29     for t = (r-1)*mptc+1:mptc*r
30         if t==1;
31             [aux,Sigma(:, :, t)] = LS( Sigma0, no,H(t,:) ',y(t,:),h0);
32             thetat(t,:) = aux.';
33         else
34             if t==(r-1)*mptc+1
35                 [aux,Sigma(:, :, t)] = LS( Sigma0, no,H(t,:) ',y(t,:),thetat(t-1,:).');
36                 thetat(t,:) = aux.';
37             else
38                 if (t>=(r-1)*mptc+1+pilot)
39                     thetat(t,:) = thetat(t-1,:);
40                     Sigma(:, :, t) = Sigma(:, :, t-1);
41                 else
42                     [aux,Sigma(:, :, t)] = LS(Sigma(:, :, t-1), no, H(t,:) ',y(t,:),thetat(t-1,:).');
43                     thetat(t,:) = aux.';
44                 end
45             end
46             Sig(t,:) = diag(Sigma(:, :, t)).';
47         end
48
49
50
51
52
53 end
54
55     for t = (R)*mptc+1:N
56         if t==(R)*mptc+1
57             [aux,Sigma(:, :, t)] = LS( Sigma0, no,H(t,:) ',y(t,:),thetat(t-1,:).');
58             thetat(t,:) = aux.';
59         else
60             [aux,Sigma(:, :, t)] = LS(Sigma(:, :, t-1), no, H(t,:) ',y(t,:),thetat(t-1,:).');
61             thetat(t,:) = aux.';
62         end
63     end
64
65
66 end
67 end
68
69 function [theta,Sigman,K] = LS( Sigmaant, varn, h,z,thetaant)
70 %Una iteracion del LS
71 K = ganancia(Sigmaant, varn, h);
72 Sigman = Sigma(h,Sigmaant, K);
73 theta= thetan(thetaant,K,h,z);
74
75 end

```

```

76
77 function [ K ] = ganancia( Sigmaant, varn, h)
78 % obtiene la ganancia del estimador lineal
79 K =Sigmaant *h/(varn +h'*Sigmaant*h);
80
81
82 end
83
84 function [ Sigman ] = Sigma( h,Sigmaant, K)
85 %Obtiene la matriz sigma del estimador secuencial (C en el informe)
86 p = length(h);
87 Sigman = (eye(p) - K*h')*Sigmaant;
88 end
89
90 function [ theta ] = thetan( thetaant,K,h, z)
91 %Obtiene la estimacion de h del tiempo actual
92 theta= thetaant + K*(z-h'*thetaant);
93
94 end

```

A.2. Simulaciones

Código Fuente: fsksim.m

```

1 function [ber, numBits] = bertooltemplate(EbNo, maxNumErrs, maxNumBits)
2 % Import Java class for BERTool.
3 import com.mathworks.toolbox.comm.BERTool;
4
5 totErr = 0; % Numero de errores observados
6 numBits = 0; % Numero de bits procesados
7
8 %Parametros
9 siglen = 1000; % Numero de bits en cada realizacion
10 M = 2; % Numero de simbolos.
11 k = log2(M);
12 ntrials = 0; %
13
14
15 freqsep = 10000; % Frecuencia de muestreo del modulador
16 nsamp = 16; %Muestras por simbolo a la salida del fsk
17 Fs = 2*freqsep; % Frecuencia de muestreo a la salida del fsk
18 snr =EbNo+10*log10(k)-10*log10(nsamp);
19
20
21
22
23 while((totErr < maxNumErrs) && (numBits < maxNumBits))
24
25
26     if (BERTool.getSimulationStop)
27         break;
28     end
29
30     msg = randi([0 1],siglen,1); % generar mensaje.
31
32
33 txsig = fskmod (msg,M,freqsep,nsamp,Fs); % Modulacion.
34 rxsig = awgn(txsig, snr, 'measured',[],'dB'); % Agrega ruido.
35 decodmsg = fskdemod(rxsig,M,freqsep,nsamp,Fs); % Demodulacion.

```

```

36 newerrs = biterr(msg,decodmsg); % Errores en esta realizacion
37 ntrials = ntrials + 1; % numero de realizaciones a la fecha.
38
39 % Numero total de errores a la fecha
40 totErr = totErr + newerrs;
41
42 % Numero total de bis procesados a la fecha
43 numBits = ntrials * siglen;
44
45
46 end
47
48 % Calcula BER.
49 ber = totErr/numBits;

```

Código Fuente: riciansim.m

```

1 function [ber, numBits] = bertooltemplate(EbNo, maxNumErrs, maxNumBits)
2 % Import Java class for BERTool.
3 import com.mathworks.toolbox.comm.BERTool;
4
5 % Initialize variables related to exit criteria.
6 totErr = 0; % Number of errors observed
7 numBits = 0; % Number of bits processed
8
9 %—— Set up parameters. ——
10 % Set up initial parameters.
11 siglen = 1000; % Number of bits in each trial
12 M = 2; % DBPSK is binary.
13 k = log2(M);
14 ntrials = 0; % Number of passes through the loop
15
16
17 freqsep = 10000; % por Bell 202 (2200Hz para 1 y 1200Hz para 0)
18 nsamp = 16; %muestras por simbolo a la salida del fsk
19 Fs = 2*freqsep; % Frecuencia de muestreo a la salida del fsk
20 snr =EbNo+10*log10(k)-10*log10(nsamp);
21
22 dop = 10; %doppler shift
23 ts = 1/Fs;
24
25
26
27 tau = [0 10^-8 10^-7 10^-6];
28 a = [0 -100 -100 -100];
29 K=9;
30
31 chan = ricianchan(ts,dop,K,tau,a);
32 chan.DopplerSpectrum = doppler.flat;
33
34 while((totErr < maxNumErrs) && (numBits < maxNumBits))
35
36
37 if (BERTool.getSimulationStop)
38 break;
39 end
40
41
42 msg = randi([0 1],siglen,1); % generar mensaje.
43
44 txsig = fskmod(msg,M,freqsep,nsamp,Fs); % Modula
45 fadedsig = filter(chan,txsig); %Pasar la senial modulada por el canal

```

```

46 rxsig = awgn(fadedsig, snr, 'measured',[],'dB'); % Agregar ruido.
47 decodmsg = fskdemod(rxsig,M,freqsep,nsamp,Fs); % Demodula
48 newerrs = biterr(msg,decodmsg); % Errores de esta realizacion
49 ntrials = ntrials + 1; % Numero de realizaciones a la fecha
50
51 % Numero total de errores a la fecha
52 totErr = totErr + newerrs;
53
54 % Numero total de bits procesados a la fecha
55 numBits = ntrials * siglen;
56
57
58 end
59 % Calcular BER
60 ber = totErr/numBits;

```

Código Fuente: algoritlin.m

```

1 %%Algoritmo de estimacion lineal
2
3 M = 2; % numero de simbolos
4 bits = 5000; %numero total de bits que se utilizaran
5 realizaciones = 100; %numero de realizaciones que tendra el algoritmo
6 l = 3; %rango
7 disp('creando bits');
8 tx = randi([0 1],bits,1); % bits aleatorios a la entrada
9
10 %% FSK
11 freqsep = 10000; % frecuencia de muestreo del fsk
12 nsamp = 16; %muestras por simbolo a la salida del fsk
13 Fs = 2*freqsep; % Frecuencia de muestreo a la salida del fsk
14 %%
15 dop = 10; %doppler shift
16 ts = 1/Fs; % tiempo de muestreo
17 %%
18 EbNo=5;
19 k = log2(M);
20 SNR =EbNo+10*log10(k)-10*log10(nsamp);
21 %%
22 Tc = 0.25/dop; % tiempo de coherencia
23 mptc = fix(Fs*Tc); % numero de muestras por tiempo de coherencia
24 R= fix(nsamp*bits/mptc); % veces que cabe el tiempo de coherencia en todo el tiempo de la ←
    simulacion (en el informe definido como numero de intervalos c)
25 %% Simular canal Riciano
26 tau = [0 10^-8 10^-7 10^-6]; %vector de retrasos en canal riciano
27 a = [0 -100 -100 -100]; %vector de ganancias en canal riciano en dB
28 K=9; %factor K de canal riciano
29
30 disp('creando canal');
31
32 chan = ricianchan(ts,dop,K,tau,a); %crear canal Riciano en matlab
33
34 chan.DopplerSpectrum = doppler.flat;
35 chan.StoreHistory = 1;
36 %% Simular modulador
37 fskSig = fskmod (tx,M,freqsep,nsamp,Fs); % modular la senial
38
39
40 %%
41 disp('Calculando H');
42
43 H = creaH2(fskSig,l); %calcular H

```

```

44
45
46
47 faded = filter(chan,fskSig); % pasa senial por el canal
48
49 ateo= chan.PathGains; % obtiene vectores de ganancia teoricas del canal
50 hteo = a2h(atteo,tau,l,Fs); % obtiene parametros h teoricos del canal
51
52 hmedia = zeros(nsamp*bits,l,21); %se guardara la media de los parametros h para cada nivel ←
    de ruido
53 hvar = zeros(nsamp*bits,l,21); %se guardara la media de la varianza de los parametros h para←
    cada nivel de ruido
54 pilot = [mptc 0.5*mptc 0.25*mptc 0.1*mptc 0.05*mptc 0.01*mptc]; %secuencias piloto que se ←
    utilizaran
55 for p = 1:6; %pilotos
56 for Ebno =0:1:20 %realizaciones
57     err = 0;
58     err2 = 0;
59
60     hest = zeros(bits*nsamp,l,realizaciones); %se guardara el valor de los parametros h para←
        cada realizacion
61 for n = 1:realizaciones
62
63     k = log2(M); %valor que se utilizara para la conversion entre SNR y ebno
64     SNR =Ebno+10*log10(k)-10*log10(nsamp); %conversion de ebno a snr
65
66     disp(strcat(int2str(p),'.','int2str(Ebno),'.','int2str(n)));
67     fadedSig = faded(H,hteo);
68     rxSig = awgn(fadedSig, SNR); %awgn para la senial
69     [ hest(:,:,n),covh] = estimador( H,rxSig,mptc,R,pilot(p),n); %utiliza el estimador lineal ←
        para obtener una estimacion de los parametros h
70
71 end
72 hmedia(:,:,Ebno+1) = mean(hest,3); %calcula la media de las estimaciones para el nivel de ruido ←
    que se este
73 hvar(:,:,Ebno+1) = std(hest,0,3); %calcular la media de la varianza de las estimaciones para el←
    nivel de ruido que se este
74
75 end
76 save(strcat(int2str(p),'.mat'),'hmedia','hvar','covh') %guarda las variables de interes
77 %% Borra variables
78 %se borran las variables antes de seguir con la siguiente secuencia piloto ,
79 %se hace debido a que el espacio de memoria asignado a matlab en la RAM no
80 %es suficiente como para mantener todas las variables de cada secuencia
81 %piloto .
82 clear hest;
83 clear hmedia;
84 clear hvar;
85 clear rxSig;
86 clear fadedSig;
87
88
89 end
90
91
92
93 %%%
94
95
96
97 %%%
98
99 energial = mean (abs(fskSig).*abs(fskSig));
100 energia2 = mean (abs(rxSig).*abs(rxSig));

```

Código Fuente: matrizcorrlin.m

```

1
2 for x =1:l % iterar en secuencias piloto
3 M = 2; % numero de simbolos
4 bits = 5000; %numero total de bits
5 realizaciones = 100; %realizaciones de la simulacion
6 l = 3; %ango
7 disp('creando bits');
8 tx = randi([0 1],bits,1); % bits aleatorios a la entrada
9
10 pilot = [500 250 125 25 5]; %secuencias piloto en utilizacion
11
12 %% Datos del modulador
13 freqsep = 10000; % frecuencia de separacion del FSK
14 nsamp = 16; %muestras por simbolo a la salida del FSK
15 Fs = 2*freqsep; % Frecuencia de muestreo a la salida del FSK
16 %% Datos de doppler
17 dop = 10; %doppler spread
18 ts = 1/Fs; %tiempo de muestreo
19
20 EbNo=5;
21 %k = log2(M);
22 %SNR =EbNo+10*log10(k)-10*log10(nsamp);
23
24 Tc = 0.25/dop; %tiempo de coherencia
25 mptc = fix(Fs*Tc); %muestras por tiempo de coherencia
26 R= fix(nsamp*bits/mptc); %numero de veces que cabe el tiempo de coherencia en el tiempo total de←
    simulacion
27 %% Simulacion del canal
28 tau = [0 10-8 10-7 10-6]; %vector de retrasos del canal (en segundos)
29 a = [0 -100 -100 -100]; %vector de amplitudes del canal (en dB)
30 K=9; %factor -k del canal
31
32 disp('creando canal');
33 chan = ricianchan(ts,dop,K,tau,a);
34
35 chan.DopplerSpectrum = doppler.flat;
36 chan.StoreHistory = 1;
37 fskSig = fskmod(tx,M,freqsep,nsamp,Fs); % modular la senial
38
39 %tau2 = [0 (10-5)/2 (10-5)/4 (10-5)/3];
40 %tau2 = tau;
41 m = length(tau);
42 %%
43 disp('Calculando H');
44
45
46
47 H = creaH2(fskSig,l); %calcula matriz de observaciones H
48 ateo = zeros(nsamp*bits,m,realizaciones); %
49 hteo = zeros(realizaciones,nsamp*bits); %
50 hestim = zeros(realizaciones,nsamp*bits); %espacio de memoria donde se guardaran las ←
    estimaciones h de cada realizacion
51 for Ebno = 15:15 % iterar en el ruido (en este caso solo 15 de ebno, 2 de SNR)
52
53
54 k = log2(M);
55 SNR =Ebno+10*log10(k)-10*log10(nsamp); %conversion de Ebno a SNR
56
57 for n = 1:realizaciones %iterar en las realizaciones
58 disp(n);
59
60 fadedSig = filter(chan,fskSig); % pasar senial por el canal

```

```

61     rxSig = awgn(fadedSig, SNR, 'measured',[],'dB'); %awgn para la senial
62     %rxSig =fadedSig; %sin awgn
63     %rxS(n,1)= rxSig;
64
65     ateo(:, :, n)= chan.PathGains; %obtiene los parametros a teoricos del canal
66     aux = a2h(ateo(:, :, n),tau,l,Fs);
67     hteo(n, :) = conj(aux(:,1)'); %obtiene los parametros h teoricos del canal
68
69     [hmed,covh] = estimador( H,rxSig,mptc,R,pilot(x),n); %obtiene los parametros h estimados ←
        linealmente
70     hestim(n, :) = conj(hmed(:,2)');
71
72 end
73 end
74 %%Obtencion de la matriz de correlaciones
75 numero = 50; %salto de muestras (debido a que la memoria no es suficiente como para calcular la ←
        matriz de correlaciones con todas las muestras)
76 hteo2 = zeros(realizaciones,fix(nsamp*bits/numero));
77 hestim2 = zeros(realizaciones,fix(nsamp*bits/numero));
78 for r = 1:l:fix(nsamp*bits/numero)
79     hteo2(:,r) = hteo(:,(r-1)*numero+1); %nuevos parametros hteo pero con menos muestras
80     hestim(:,r) = hestim(:,(r-1)*numero+1); %nuevos parametros h estimados pero con menos ←
        muestras
81 end
82 corrtteo = corr(hteo2); %matriz de correlacion teorica
83 correstim = corr(hestim2); %matriz de correlacion estimada
84
85 %% Graficar correlaciones
86 q = 0:50:(nsamp*bits)-50;
87 %warteo = diag(cov(hteo2));
88 %warestim = diag(cov(hestim2));
89 colores = ['g','r','c','m','k','y'];
90 q = q*ts;
91 plot(q(1:800),corrtteo(1,1:800),'—',q(1:800),correstim(1,1:800),'-')
92 grid on;
93 xlabel('Tiempo[s]')
94 ylabel('corr(h_l[0],h_l[n])')
95 legend('correlacion canal teorico','correlacion canal estimado');
96 saveas(gcf, strcat('correlacion tipo de muestra',int2str(x)), 'pdf');
97 save(strcat('corr',int2str(x),'.mat'),'corrtteo','correstim')
98 clear;
99 end

```

Código Fuente: algoritsec.m

```

1 %Algoritmo de estimacion secuencial.
2 pilot =[500 250 125 50 25 5]; % largo de secuencias piloto que se utilizaran
3 for pil = 1:l:6 %iterar en todas las secuencias piloto
4
5 M = 2; % numero de simbolos
6 bits = 5000; %numero de bits
7 realizaciones = 100; %numero de realizacion por cada nivel de ruido
8 l =3; %rango
9 disp('creando bits');
10 tx = randi([0 1],bits,1); % bits aleatorios a la entrada
11
12 %% Modulador FSK
13 freqsep = 10000; % frecuencia de separacion de modulador FSK
14 nsamp = 16; %muestras por simbolo a la salida del FSK
15 Fs = 2*freqsep; % Frecuencia de muestreo a la salida del FSK
16 %%
17 dop = 10; %doppler shift

```



```

18 | ts = 1/Fs; %periodo de muestreo
19 | %%% %%% %%% %%% %%% %%%
20 | EbNo=5;
21 | k = log2(M);
22 | SNR =EbNo+10*log10(k)-10*log10(nsamp);
23 | %%% %%% %%% %%% %%% %%%
24 |
25 | Tc = 0.25/dop; %Tiempo de coherencia
26 | mptc = fix(Fs*Tc); %muestras por el tiempo de coherencia
27 | R= fix(nsamp*bits/mptc); % numeros de veces que cabe el tiempo de coherencia en el tiempo total ←
    |     de simulacion (el numero total de c en el informe)
28 |
29 | %% Simular el canal Riciano
30 | tau = [0 10^-8 10^-7 10^-6]; %vector de retrasos para el canal Riciano (en segundos)
31 | a = [0 -100 -100 -100]; %vector de ganancia para el canal Riciano (en dB)
32 | K=9; %factor-k del canal Riciano
33 |
34 | disp('creando canal');
35 | chan = ricianchan(ts,dop,K,tau,a); %crea canal en matlab
36 |
37 | chan.DopplerSpectrum = doppler.flat;
38 | chan.StoreHistory = 1;
39 | fskSig = fskmod(tx,M,freqsep,nsamp,Fs); % modular la senial en FSK no-coherente
40 |
41 | %tau2 = tau;
42 | %m = length(tau);
43 |
44 | disp('Calculando H');
45 | H = creaH2(fskSig,l);
46 |
47 |
48 |
49 | faded = filter(chan,fskSig); % pasar senial el canal
50 |
51 | ateo= chan.PathGains; %obtiene el vector de ganancia teorica del canal (en cada instante nTs←
    | )
52 | hteo = a2h(ateo,tau,l,Fs); %obtiene los parametro h teoricos del canal (en cada instante nTs←
    | )
53 | clear ateo;
54 | hmedia = zeros(nsamp*bits,l,21);
55 | %%
56 | for Ebno =0:1:20 %iterar en el ruido
57 |
58 |     K = zeros(l,l,nsamp*bits);
59 |     fadedSig = fadeds(H,hteo);
60 |     disp(strcat(int2str(pil),'.',int2str(Ebno)));
61 |     hest= zeros(nsamp*bits,l,realizaciones);
62 |     Sigma = zeros(nsamp*bits,l);
63 | %%
64 | for n = 1:realizaciones %iterar en las realizaciones
65 |
66 |     k = log2(M); %valor que se utilizara para convertir de Ebno a SNR
67 |     SNR =Ebno+10*log10(k)-10*log10(nsamp);
68 |     snr= 10^(SNR/10);%conversion de Ebno a SNR
69 |     rxSig = awgn(fadedSig, SNR); %awgn para la senial
70 |
71 |     [ hest(:, :, n),Sig] = estimadorseq( H,rxSig,mptc,R,1,pilot(pil)); %estimador secuencial de h ←
    |     para cada realizacion
72 |     Sigma = Sigma +Sig; %matriz de covarianza acumulada
73 |     clear Sig;
74 |     clear rxSig
75 | end
76 |
77 | hmedia(:, :, Ebno+1) = mean(hest,3); %media de h en las realizaciones

```

```

78 hvar(:, :, Ebno+1) = std(hest,0,3); %media de la varianza de h en las realizaciones
79 Sigmam(:, :, Ebno+1) = Sigma/realizaciones; %media de matriz de covarianza
80
81 end
82 %mse = mean(abs(hest-hteo).*abs(hest-hteo),1)';
83
84 %a = 1:1:bits*nsamp;
85 %%%
86 %numero = 80000;
87 %plot(t(1:numero),real(hest(1:numero)),'-',t(1:numero),real(hteo(1:numero,1)),'--');
88
89 save(strcat(int2str(pil),'.mat'),'hmedia','hteo','hvar','H','Sigmam');
90 clearvars -except pilot pil;
91 end

```

Código Fuente: matrizcorrsec.m

```

1
2 for x =1:1 % iterar en secuencias piloto
3 M = 2; % numero de simbolos
4 bits = 5000; %numero total de bits
5 realizaciones = 100; %realizaciones de la simulacion
6 l = 3; %ango
7 disp('creando bits');
8 tx = randi([0 1],bits,1); % bits aleatorios a la entrada
9
10 pilot = [500 250 125 25 5]; %secuencias piloto en utilizacion
11 %% Datos del modulador
12 freqsep = 10000; % frecuencia de separacion del FSK
13 nsamp = 16; %muestras por simbolo a la salida del FSK
14 Fs = 2*freqsep; % Frecuencia de muestreo a la salida del FSK
15
16 %% Datos de doppler
17 dop = 10; %doppler spread
18 ts = 1/Fs; %tiempo de muestreo
19
20
21 %ebNo=5;
22 %k = log2(M);
23 %SNR =ebNo+10*log10(k)-10*log10(nsamp);
24
25 Tc = 0.25/dop; %tiempo de coherencia
26 mptc = fix(Fs*Tc); %muestras por tiempo de coherencia
27 R = fix(nsamp*bits/mptc); %numero de veces que cabe el tiempo de coherencia en el tiempo total de←
    simulacion
28 %% Simulacion del canal
29 tau = [0 10^-8 10^-7 10^-6]; %vector de retrasos del canal (en segundos)
30 a = [0 -100 -100 -100]; %vector de amplitudes del canal (en dB)
31 K=9; %factor-k del canal
32
33 disp('creando canal');
34 chan = ricianchan(ts,dop,K,tau,a);
35
36 chan.DopplerSpectrum = doppler.flat;
37 chan.StoreHistory = 1;
38 fskSig = fskmod(tx,M,freqsep,nsamp,Fs); % modular la senial
39
40 %tau2 = [0 (10^-5)/2 (10^5)/4 (10^-5)/3];
41 %tau2 = tau;
42 %m = length(tau2);
43
44 %est = zeros(m,realizaciones);

```

```

45 %%
46 disp('Calculando H');
47
48
49
50 H = creaH2(fskSig,1);%calcula matriz de observaciones H
51
52 for ebno =15:15
53
54
55     k = log2(M);
56     SNR =ebno+10*log10(k)-10*log10(nsamp);%conversion de ebno a SNR
57     hteo = zeros(realizaciones,nsamp*bits);
58     hestim = zeros(realizaciones,nsamp*bits);
59
60 for n = 1:realizaciones
61 disp(strcat( int2str(ebno),int2str(n)));
62
63     fadedSig = filter(chan,fskSig);    % pasar senial el canal
64
65     %rxSig =fadedSig; %sin awgn
66     %rxS(n,1)= rxSig;
67
68     ateo= chan.PathGains;%obtiene los parametros a teoricos del canal
69     auxteo = a2h(ateo,tau,1,Fs);
70     hteo(n,:) = auxteo(:,1).'; %obtiene los parametros h teoricos del canal
71
72     fadedSig =fadedS( H,auxteo);% pasar senial el canal (funcion propia)
73     rxSig = awgn(fadedSig, SNR, 'measured',[],'dB');    %awgn para la senial
74     no =1;
75
76     [ aux,Sig] = estimadorseq( H,rxSig,mptc,R,no,pilot(x));%obtiene los parametros h estimados ←
77         secuencialmente
78     hestim(n,:) = aux(:,1).';
79
80 end
81
82 end
83 %%Obtencion de la matriz de correlaciones
84 numero = 50; %salto de muestras (debido a que la memoria no es suficiente como para calcular la ←
85     matriz de correlaciones con todas las muestras)
86 hteo2 = zeros(realizaciones,fix(nsamp*bits/numero));
87 hestim2 = zeros(realizaciones,fix(nsamp*bits/numero));
88 for r = 1:1:fix(nsamp*bits/numero)
89     hteo2(:,r) = hteo(:,(r-1)*numero+1); %nuevos parametros hteo pero con menos muestras
90     hestim2(:,r) = hestim(:,(r-1)*numero+1); %nuevos parametros h estimados pero con menos ←
91     muestras
92 end
93
94 corrtteo = corr(hteo2); %matriz de correlacion teorica
95 correstim = corr(hestim2); %matriz de correlacion estimada
96
97 %% Graficar correlaciones
98 q = 0:50:(nsamp*bits)-50;
99 %warteo = diag(cov(hteo2));
100 %warestim = diag(cov(hestim2));
101 colores = ['g','r','c','m','k','y'];
102 q = q*ts;
103 plot(q(1:800),corrtteo(1,1:800),q(1:800),correstim(1,1:800),colores(x))
104 grid on;
105 xlabel('Tiempo[s]')
106 ylabel('corr(h_1[0],h_1[n])')
107 legend('correlacion canal teorico','correlacion canal estimado');
108 saveas(gcf,strcat('correlacion tipo de muestra',int2str(x)), 'pdf');

```

```
106 | save(strcat('corr',int2str(x),'.mat'),'corroteo','correstim')
107 | clear;
108 | end
109 | end
110 |
111 | %correlacion = corr(hteo);
```

