



UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA



Implementación de un transmisor de ISDB-T abierto bajo el paradigma de Radio Definida por Software

MEMORIA DE PROYECTO PRESENTADA A LA FACULTAD DE
INGENIERÍA DE LA UNIVERSIDAD DE LA REPÚBLICA POR

Javier Hernández, Santiago Castro

EN CUMPLIMIENTO PARCIAL DE LOS REQUERIMIENTOS
PARA LA OBTENCIÓN DEL TÍTULO DE
INGENIERO ELECTRICISTA.

TUTOR

Dr. Federico La Rocca Universidad de la República
M.Sc. Pablo Flores Guridi Universidad de la República

TRIBUNAL

M.Sc. Alicia Fernández Universidad de la República
Dr. Víctor González-Barbone Universidad de la República
Dr. Rafael Sotelo Universidad de la República

Montevideo
lunes 1 octubre, 2018

*Implementación de un transmisor de ISDB-T abierto bajo el paradigma de Radio
Definida por Software*, Javier Hernández, Santiago Castro.

Esta tesis fue preparada en L^AT_EX usando la clase iietesis (v1.1).
Contiene un total de 59 páginas.
Compilada el lunes 1 octubre, 2018.
<http://iie.fing.edu.uy/>

Resumen

Han pasado 7 años desde que Uruguay adoptó el estándar ISDB-T como la norma nacional para la transmisión de televisión digital. Encontramos que al día de hoy, el estado de la red dista bastante de lo proyectado cuando se firmó el decreto. El acceso a la información técnica se ve dificultado por la poca cantidad de documentos alusivos al alcance. Además, para la formación de técnicos en el área, es indispensable contar con hardware muy costoso y con distintos softwares propietarios, de difícil acceso tanto para los interesados.

Nos proponemos en este trabajo aportar desde nuestro lugar en el rubro, más información sobre la norma y su implementación, respaldando los conocimientos teóricos adquiridos con una solución técnica accesible para todos.

Es por eso que, complementando el trabajo iniciado por el grupo ARTES de la Facultad de Ingeniería en 2016 con el proyecto gr-isdbt, presentamos un transmisor de televisión digital, implementado completamente en el formato de código abierto y amparado en el paradigma de Radio Definida por Software.

De este modo, cualquier interesado en entender la norma será capaz de poner en funcionamiento un sistema de transmisión de televisión digital a pequeña escala. La conjunción entre gr-isdbt y gr-isdbt-tx, permite trabajar con el sistema completo de punta a punta, teniendo acceso absoluto a toda la información en transmisión, en cualquier punto de la cadena.

Esta página ha sido intencionalmente dejada en blanco.

Tabla de contenidos

Resumen	I
1. Introducción	1
2. Fundamento Teórico	5
2.1. Modelado del canal	6
2.1.1. Canal de Gauss	6
2.1.2. Canal de Rice	6
2.1.3. Canal de Rayleigh	7
2.2. Estrategias para mitigar los efectos del canal	7
2.2.1. Códigos de detección y corrección de errores	7
2.2.2. Códigos Cíclicos	8
2.3. Modulación OFDM	10
2.4. MPEG y sus Estandares	11
2.4.1. MPEG 2 Transport Stream	11
2.4.2. Tablas PMT	12
2.4.3. Tablas PAT	12
2.4.4. Paquetes Nulos	12
3. El Sistema de Televisión Digital Terrestre ISDB-T	13
3.1. BTS como fuente de datos	14
3.2. Robustecimiento frente a las no idealidades del canal	16
3.3. Formacion de los cuadros OFDM	17
3.4. Las portadoras y la modulacion	17
3.5. La puesta en el aire de la señal	18
3.5.1. Necesidad del prefijo cíclico	18
3.5.2. La potencia requerida para transmitir	20
4. Radio definida por Software	21
4.1. GNU Radio	21
4.1.1. Flowgraphs	22
4.1.2. Bloques	22
4.2. Hardware	23
4.2.1. USRP B100	24

Tabla de contenidos

5. Un transmisor ISDBT implementado en GNU Radio	27
5.1. El flujo de datos en GNU Radio	27
5.2. Obtencion de los TSP por capa	28
5.3. Codificaciones de Canal	29
5.3.1. Reed Solomon	29
5.3.2. Viterbi	29
5.4. La modulacion	30
5.5. El uso de los entrelazamientos	32
5.5.1. Entrelazamiento frecuencial	32
5.5.2. Entrelazamiento temporal	32
5.6. Entrelazamiento de bits	34
5.7. Formacion de cuadros OFDM	36
5.8. La transformada de Fourier	37
5.9. El prefijo cíclico	38
5.10. La transmision desde USRP	39
6. Evaluación del sistema	41
6.1. Pruebas sobre gr-isdbt	41
6.2. Pruebas sobre televisores comerciales	43
7. Conclusiones y trabajo a futuro	45
Referencias	47
Índice de tablas	48
Índice de figuras	50

Capítulo 1

Introducción

En Mayo de 2001, la ARIB (Association of Radio Industries and Businesses) [1] presentó la primera versión de su estándar para la transmisión de televisión digital. Coloquialmente denominada “Norma Japonesa de Televisión Digital”, académicamente ISDB-T, por sus siglas en ingles provenientes de *Integrated Services Digital Broadcasting, Terrestrial*. La norma sintetiza un conjunto de requerimientos técnicos para la utilización eficiente del espectro radioeléctrico para la transmisión de datos multimedia, con la colaboración y el aval de todos los actores de la industria, como proveedores de tecnología, compañías de broadcasting, investigadores del área, entre otros.

Esta norma basa muchos de sus conceptos en la norma DVB-T, publicada por primera vez en el año 1997 por la organización europea DVB (Digital Video Broadcasting). La posterioridad de ISDB-T con respecto a ésta permitió que se robustecieran algunos de los aspectos más criticados de la norma europea, resultando en un estándar mas robusto para la transmisión. Discutiremos algunos de estos aspectos mas adelante, cuando se presente la norma en el capítulo 3.

Actualmente, existen en el mundo cuatro grandes estándares comerciales. Además de ISDB-T y DVB-T, están la norma china DTMB (Digital Terrestrial Multimedia Broadcast) y la norma norteamericana de la ATSC (Advanced Television Systems Comitee). La elección de que norma adoptar por parte de los gobiernos nacionales radica exclusivamente en decisiones políticas, cuyo análisis y discusión escapan a los objetivos de este documento.

En Uruguay, al igual que en gran parte de Latinoamérica, se adoptó en 2011 una versión de ISDB-T denominada ISDB-T International, la cual es a grandes rasgos idéntica a la primera, salvo por algunos cambios menores como el cambio en la codificación de fuente (pasa del estándar MPEG-2 a MPEG-4) y la elección de otro estándar de interactividad (se cambia de BML a Ginga).

Luego de la adopción del estándar, se fijó para el año 2015 como la fecha limite para el denominado “apagón analógico”, fecha en la cual se dejaría de transmitir televisión por vías analógicas, pasando exclusivamente a medios digitales, liberando los espectros asignados para los canales de TV abierta para otros fines.

Durante la implementación del marco legal de la nueva norma de televisión digital se entregaron 22 licencias para transmisión de contenidos bajo la norma

Capítulo 1. Introducción

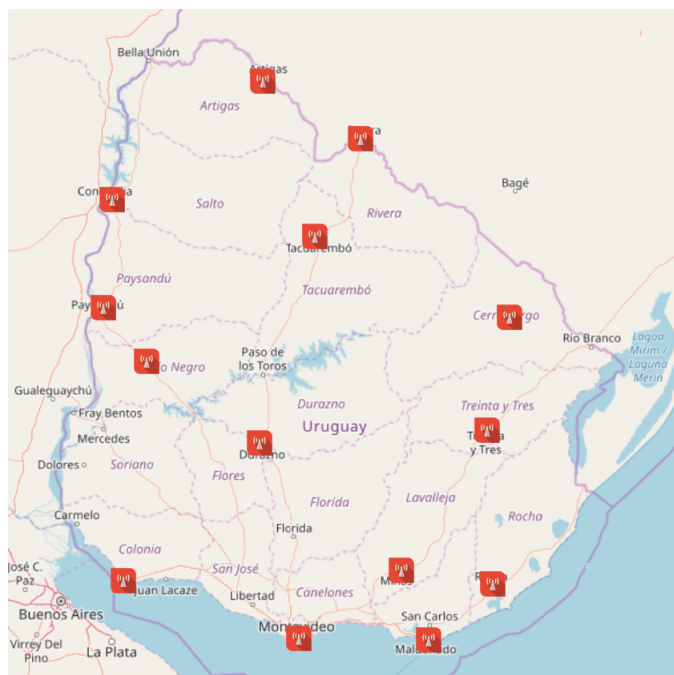


Figura 1.1: Distribución de las estaciones transmisoras de TV digital en el Uruguay.

ISDB-T International (en adelante ISDB-T por simplicidad). Al día de hoy, tres años después de la fecha límite para el apagón, solo algunos de los actores del rubro están brindando el servicio de forma adecuada y llama la atención la baja participación del sector comercial en la transformación analógico-digital.

Relevamos la situación de las estaciones de transmisión de televisión digital a lo largo del país, y hemos encontrado que la cobertura abarca solo una parcialidad del territorio nacional, existiendo incluso departamentos del interior del país que aun no tienen cobertura en su totalidad.

La situación de los consumidores del servicio también dista de la ideal, pues la televisión analógica sigue siendo la mayor puerta de acceso al medio. La Encuesta Continua de Hogares del Instituto Nacional de Estadística [4], cuyos indicadores son una muestra representativa de la situación de todos los hogares del país, dió a conocer que en el año 2017 solamente el 47 % de los hogares encuestados tienen recepción de TV digital abierta. Tanto es así, que el mencionado apagón analógico fue pospuesto por tiempo indeterminado. El alto costo del recambio de equipamiento, y posturas sobre la democratización del acceso a la información para personas de bajos recursos, fundamentan estas decisiones.

Durante el auge de la TVD, en los momentos posteriores a la adopción de la norma, surgió la necesidad de generar documentación técnica para la implementación de la misma. Mediante una colaboración entre la URSEC y la Facultad de Ingeniería, se desarrolló el proyecto gr-isdbt [5]. Pablo Flores, Gabriel Gómez, Víctor González-Barbone y Federico Larroca presentaron un receptor de televisión digital, de código abierto y bajo el paradigma de radio definida por software (SDR, paradigma de transmisión que explicaremos en detalle mas adelante).

Nuestro proyecto complementa el trabajo iniciado en *gr-isdbt*, pues con la implementación del transmisor ISDBT, se completa el sistema de transmisión de televisión digital de punta a punta. Ambos trabajos en conjunto permiten visualizar y ayudan a comprender el funcionamiento del sistema presentado en la norma, teniendo acceso completo a todo lo que sucede dentro del mismo, en cualquier punto de la cadena de transmisión y recepción.

Además, genera la posibilidad de recrear una planta de transmisión de televisión a muy bajo costo, permitiendo su implementación tanto en el hogar por entusiastas, en el aula por docentes o en la industria, por técnicos, lo cual puede colaborar con el mejoramiento de la calidad del servicio actual. A su vez, puede servir incluso como ejemplo para algunos de los cursos de la Facultad, generalmente catalogados por el estudiantado como muy teóricos y con poco alcance práctico.

Implementar un transmisor de televisión digital no es una tarea sencilla. El primer problema a enfrentar es el acceso a la información técnica. Existe poca documentación generada en el país para cumplir con las condiciones técnicas de un sistema complejo y que, además, ya lleva 7 años de vigencia como oficial. La norma presentada por la ARIB deja varias zonas grises, asume por conocidos conceptos clave, y no se explora más de lo necesario en cuestiones de fondo.

También existen fuertes limitaciones económicas para hacerse con software o hardware comercial que resuelvan incluso algunas de las funcionalidades más básicas que exige la norma.

Esta tesis intenta suplir con ambas, complementando el trabajo iniciado por el grupo ARTES con el receptor *gr-isdbt*. Se desarrolló a lo largo de este proyecto, un transmisor de televisión digital que cumple con las condiciones establecidas en la norma, y cuyas señales son decodificables por los televisores comerciales homologados por el LATU. El resultado es la posibilidad de implementar a pequeña escala y por un costo considerablemente menor, un sistema funcional de transmisión de TVD. Además, presentamos en este documento, un análisis de la norma y explicamos de que forma se resolvieron los problemas de implementación y de las zonas “poco claras” que se mencionaron antes.

Contar con el trabajo presentado en *gr-isdbt* fue de una ayuda mayúscula, ya que disponer del código completo del receptor ayudó a comprender, sintetizar y testear los conceptos teóricos vertidos en la norma, lo que fue fundamental para la comprensión de las funcionalidades que sería necesario implementar para poder transmitir.

Para este grupo de trabajo es importante destacar lo valioso de la existencia de proyectos de código abierto. Incontables veces encontramos en la comunidad puntos de vista, ideas y hasta algoritmos para resolver los problemas encontrados en el camino. Es por eso que esperamos poder contribuir con ella, poniendo a disposición de cualquier persona el transmisor *gr-isdbt-tx*, para que continúen con el trabajo de aprendizaje y la optimización del mismo por técnicos y estudiantes, seguramente con un mejor panorama del rubro, que el que tuvimos al implementar este proyecto.

Esperamos también, mediante el desarrollo de este documento, poder contribuir con la comunidad nacional de técnicos que trabajan en el rubro, y que no

Capítulo 1. Introducción

cuentan con documentación técnica generada por y para la norma nacional, con los problemas y las particularidades que la transmisión tiene en nuestro país y no tener que abstraer de trabajos de terceros, que resolvieron problemas similares en contextos diferentes. Entendemos que en este proyecto, los conceptos desarrollados por la norma se sintetizan en órdenes básicas al procesador, y al ser de código abierto y gratuito, se democratiza el acceso a una información a la que hoy por hoy solo se accede por medio de hardware y software propietario con licencias de costos elevados.

Para esta documentación, que acompaña el código presentado para el transmisor, definimos seis capítulos en los que se explica el desarrollo del mismo. En el capítulo 2 presentamos algunos de los conceptos fundamentales de telecomunicaciones sobre los que se construye la norma. El capítulo 3 realiza un breve pasaje por los puntos clave del sistema transmisor ISDB-T, los cuales son necesarios para comprender algunos de los bloques que conforman el sistema. Para profundizar más en los mencionados conceptos, invitamos al lector a revisar la tesis de maestría de Pablo Flores, que pueden encontrar en [6]. Luego, el capítulo 4 se detiene particularmente en el concepto de radio definida por software y presenta en detalle una implementación del mismo, en particular aquel sobre el cual se desarrolló el transmisor, que es GNU Radio en conjunto con un equipo USRP. En el capítulo 5 analizamos el código generado para implementar el transmisor, explicando en cada paso los conceptos del capítulo 2 y 3 que se necesita aplicar en cada bloque, y como se extrapolaron a C++, lenguaje en el que se escribió cada uno de los bloques de procesamiento. Mas adelante, en el capítulo 6 mostramos el desempeño del transmisor como un todo, realizando las evaluaciones prácticas del mismo en función de los objetivos de este proyecto y se comentan los resultados obtenidos. Para terminar, en el capítulo 7, presentamos las conclusiones del proyecto en particular y planteamos algunos desafíos que sería interesante afrontar en un futuro.

Capítulo 2

Fundamento Teórico

Entendemos al proyecto gr-isdbt-tx como un complemento al trabajo de maestría presentado por Pablo Flores, por lo que nos parece reiterativo explicar en sumo detalle algunos de los fundamentos de telecomunicaciones sobre los que se construye la norma ISDB-T. Los mismos ya fueron explicados con gran claridad y nivel de detalle en la documentación de aquel proyecto. Quizás algunos de los lectores encuentren insuficiente el desarrollo de algunos de los conceptos incluidos en este capítulo, a ellos los invitamos a complementar la lectura de esta documentación, con la tesis de Pablo.

Es posible también, que entre los lectores se encuentren técnicos vinculados al área de sistemas y programación. Encontraran ellos, seguramente, mayor interés y profundidad de conceptos en los capítulos 4 y 5, donde detallamos el uso de GNU Radio como plataforma de SDR, y la estructura funcional de nuestro transmisor, así como los desafíos de programación a los que nos enfrentamos implementando cada uno de sus bloques.

Intentando contemplar el interés de todos los lectores, es que les presentamos en este capítulo un repaso sobre algunos de los conceptos que entendemos clave para comprender por que son necesarios algunos de los bloques del transmisor.

Comenzaremos explicando aquí como se modela el canal inalámbrico para trabajar, que problemas se enfrenta uno cuando debe diseñar un sistema de transmisión por aire y como mitigar algunos de los efectos que el canal tendrá sobre nuestra señal. Repasaremos algunos de los mas utilizados códigos de detección y corrección de errores. Luego vendrá una corta explicación sobre los sistemas OFDM, su surgimiento teórico y practico, para dar un marco de surgimiento de algunas de las funcionalidades de ISDB-T. Para terminar, explicaremos de que forma se encapsulan los datos a transmitir, para eso, repasamos algunos conceptos definidos en las normas MPEG, orientados hacia la transmisión de televisión, que ayudaran a modelar la fuente de datos del sistema.

2.1. Modelado del canal

Cuando una señal viaja a través del aire, sufrirá una serie de modificaciones a causa de muchos factores. El clima, el ruido generado por otras señales que también viajan por el mismo medio, los rebotes de la señal contra edificios, autos o cualquier objeto que se encuentre en el camino, son todos aspectos que modificarán la señal, y por lo que la señal recibida será distinta de la transmitida.

De forma teórica, se podrían plantear las ecuaciones de Maxwell, definir condiciones de borde que modelen de manera fiel los obstáculos en el camino y calcular en cada momento y en cada lugar del espacio, la atenuación que afectará nuestra señal. Pero el canal cambia constantemente, las diferentes señales que comparten el medio también están en constante transformación, y por lo tanto no sería ni práctico, ni útil seguir un camino como este para comprender el canal, mucho menos basarse en él para robustecer a la señal frente a los atenuantes que surgirán.

Es por eso que se realiza un modelado del canal. Una forma más simple de entender al medio, donde consideramos modelos puntuales de las fuentes de interferencia, para obtener una idea general del comportamiento del mismo, y modelamos el éxito en recepción mediante una función de probabilidad. Este modelo del canal podrá ser más o menos complejo, dependerá del técnico que se enfrente al problema decidir que herramientas incluir en función de los efectos que desea mitigar en su señal.

Existen tres grandes modelos de canal. Son los modelos de Gauss, Rice y Rayleigh. El primero, se utiliza fuertemente para el cálculo de enlaces punto a punto. Por su parte, el modelo de Rice, es bueno para modelar los enlaces punto a multipunto, por lo tanto se utiliza ampliamente en el mundo del broadcasting. Finalmente el modelo de Rayleigh considera los efectos de la difracción y los ecos producidos por las múltiples trayectorias.

2.1.1. Canal de Gauss

El canal de Gauss es el modelo más sencillo de canal que se trabaja. En particular, es el modelo que se utiliza para calcular los enlaces punto a punto con línea vista. Generalmente, también se presenta la hipótesis de que el resto de las interferencias del canal, pueden modelarse como un ruido aditivo Gaussiano.

2.1.2. Canal de Rice

El modelo de canal de Rice considera que un receptor estará expuesto primero hacia un haz directo, en lo que puede ser una transmisión por línea vista, pero también considera otras trayectorias, debidas a los rebotes del haz original con los distintos obstáculos del entorno. Estos rayos secundarios generalmente vendrán acompañados de un determinado delay, de dimensión considerable.

2.2. Estrategias para mitigar los efectos del canal

2.1.3. Canal de Rayleigh

En ISDB-T la transmisión en la capa A está orientada a la recepción en dispositivos móviles. De estos temas hablaremos más adelante en el capítulo 3, pero por el momento, presentar esta idea nos da pie a plantear un modelo distinto de canal, uno en el que el receptor se mueve.

El canal de Rayleigh tiene en cuenta las atenuaciones que se producen por lo que se conoce como Difracción Combinada, esto es, señales que sortean obstáculos perdiendo mucha de su potencia inicial en el proceso. Este fenómeno es interesante, puesto que puede llegar a existir transmisión, pese a que no haya línea vista.

Este modelo de canal toma mucha relevancia en ISDB-T, no solo porque la mayoría de los televisores tienen antenas internas, sino que tenemos que modelar el canal para resolver el problema de la recepción móvil. Es posible robustecer a la señal de forma tal que un teléfono móvil pueda decodificar datos multimedia en alta definición y con poca latencia en todo momento, sin perder la continuidad del servicio.

Existen una serie de estrategias, implementadas en la norma ISDB-T, orientadas a resolver estos mismos problemas.

2.2. Estrategias para mitigar los efectos del canal

2.2.1. Códigos de detección y corrección de errores

La comunicación entre emisor y receptor puede modelarse mediante el proceso de la Figura 2.1. La situación es la siguiente, una fuente emisora envía mensajes m (palabras fuente) al receptor a través de un canal de comunicación. El mensaje debe ser traducido a algún mensaje que el canal esté capacitado para enviar, estos mensajes se conocen como palabras código.

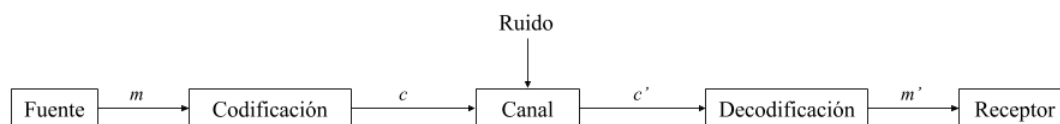


Figura 2.1: Esquema básico de codificación de canal.

Al otro lado del canal llega un mensaje codificado c' , el cual seguramente sea erróneo, pues en todo proceso real de comunicación existe ruido e imperfecciones en los canales. El mensaje es decodificado en una palabra m' , y generalmente $m' \neq m$.

Se desea que el receptor sea capaz de darse cuenta si el mensaje m' es realmente lo que se transmitió del otro lado, y más aún, poder corregirlo.

La Teoría de Códigos es un campo de la matemática aplicada que busca resolver los problemas de las etapas de codificación-decodificación y corrección, y que presenta su propia complejidad.

Capítulo 2. Fundamento Teórico

La transmisión inalámbrica de una señal la expone a diversas fuentes de ruido, con lo cual los tipos de errores generados pueden ser muy variados. Por ejemplo los errores en ráfaga, en los que un conjunto de bits consecutivos se ven alterados, son muy comunes en las comunicaciones inalámbricas. También podría suceder que el canal radioeléctrico presente distorsión en algunas portadoras en particular.

El estándar ISDB-T hace uso de distintas técnicas modernas para la protección de los datos en transmisión. De hecho para proteger los datos en los ejemplos mencionados el estándar utiliza la *dispersión de energía* y el *entrelazamiento frecuencial*. Para la comprensión del estándar y el desarrollo de *gr-isdbt-tx*, es importante conocer el funcionamiento de estas técnicas. Profundizar en estos temas escapa los objetivos de este trabajo, por lo cual los detalles técnicos se pueden encontrar en las bibliografías mencionadas.

Para asegurarse que el receptor pueda llevar a cabo satisfactoriamente la demodulación y decodificación en una transmisión jerárquica, en la cual se utilizan múltiples parámetros de transmisión, se utiliza una señal denominada Transmission and Multiplexing Configuration Control (TMCC).

Como se verá en el Capítulo XX, la TMCC junto con otras señales piloto y las señales correspondientes a la transmisión de los datos útiles, conforman el cuadro OFDM.

Al tratarse de una señal que contiene información crítica sobre la transmisión se la debe proteger fuertemente frente a los distintos tipos de errores que podría sufrir durante su transmisión.

En particular ISDB-T establece que para la TMCC se debe utilizar el código acortado (200,118) del *difference-set cyclic code* (273,191) como código corrector de errores.

2.2.2. Códigos Cíclicos

El conjunto $GF(2) \triangleq \{0, 1\}$, con las operaciones de suma " + " y producto " \times " usuales módulo 2, cumple con la propiedad de que cualquier elemento de $GF(2)$ distinto de cero tiene inverso. Esta propiedad se cumple trivialmente en este conjunto y es la condición necesaria para que $GF(2)$ sea un *Campo de Galois*. Es común encontrar que a este campo también se lo llame *campo binario* y se lo denote como \mathbb{F}_2 . Las operaciones de suma y producto definidas en $GF(2)$ son asociativas, conmutativas y distributivas, y llevan elementos de $GF(2)$ en elementos de $GF(2)$. Por esto $GF(2)$ también es un *anillo*. El conjunto de todos los polinomios con coeficientes en $GF(2)$ con las operaciones usuales de suma y producto forman un *anillo de polinomios* en $GF(2)$ y se denota como $GF(2)[x]$. Por ejemplo $g(x) = x^3 + x + 1$ es un elemento de $GF(2)[x]$.

Sea $\mathbf{c} = (c_0, c_1, \dots, c_{n-1}) \in GF(2)^n$, con $GF(2)$ tal como se describió anteriormente. Un código \mathcal{C} de bloque (n, k) se dice que es un *código cíclico* si para cada vector $\mathbf{c} = (c_0, c_1, \dots, c_{n-1}) \in \mathcal{C}$ cualquier rotación circular a la derecha de \mathbf{c} también pertenece a \mathcal{C} , es decir $(c_{n-1}, c_0, c_1, \dots, c_{n-2}) \in \mathcal{C}$. Los códigos de bloque se caracterizan por codificar mensajes de longitud fija k en *codewords* de longitud fija n , con lo cual el tamaño del mensaje original se incrementa en $n - k$. Cada *code-*

2.2. Estrategias para mitigar los efectos del canal

word del código \mathcal{C} puede ser representada en una forma polinomial de la siguiente manera:

$$c(x) = \sum_{i=0}^{n-1} c_i x^i \quad (2.1)$$

A continuación se enumera una serie de propiedades de los códigos cíclicos, en [9] se puede encontrar una demostración detallada de cada una de ellas.

- Un código cíclico es un código lineal de bloque
- Cada *codeword* se corresponde con un polinomio
- Los polinomios del código forman un *ideal* en $GF(2)[x]/(x^n - 1)$
- Para un código cíclico existe un generador $g(x)$ que es divisor de $x^n - 1$ y que puede generar todos las *codewords* $c(x) = m(x)g(x)$

Se puede probar que esto implica la existencia de una *matriz de chequeo de paridad* $\mathbb{H} \in \mathcal{M}_{(n-k) \times n}$ tal que para toda *codeword* \mathbf{c} de \mathcal{C} se cumple $\mathbf{c}\mathbb{H}^T = \mathbf{0}$.

El proceso de codificación se realiza de la siguiente manera, primero se construye el polinomio $x^{n-k}m(x)$ de grado n . Luego se divide entre el polinomio generador $g(x)$ y el resto de esa división es el polinomio de paridad $d(x)$ que se le agregará al mensaje:

$$x^{n-k}m(x) - q(x)g(x) = d(x) \quad (2.2)$$

La *codeword* se forma de la siguiente manera:

$$c(x) = x^{n-k}m(x) - d(x) = q(x)g(x) \quad (2.3)$$

Como se trata de un múltiplo de $g(x)$, entonces efectivamente es una *codeword* válida. La representación vectorial de la *codeword* queda de la siguiente manera:

$$\mathbf{c} = (-d_0, -d_1, \dots, -d_{n-k-1}, m_0, m_1, \dots, m_{k-1}) \quad (2.4)$$

En una situación en la que se recibe una palabra \mathbf{r} cuyo mensaje es \mathbf{m} y sus bits de paridad son \mathbf{d} , el procedimiento para detectar si hubo error es codificar el mensaje \mathbf{m} que se recibió con el mismo codificador utilizado por el transmisor (ambas partes deben conocer el polinomio generador), y luego comparar el \mathbf{d}' obtenido con el \mathbf{d} recibido. Si ambos difieren entonces hubo error. Por ejemplo, para un código cíclico (7, 4) con polinomio generador $g(x) = x^3 + x + 1$ se desea codificar el mensaje 1001. Los mensajes codificados tendran $n - k = 7 - 4 = 3$ bits de paridad. El mensaje en su forma polinomial queda $m(x) = 1 + x^3$. Los bits de paridad se obtienen calculando el resto de la division $x^{(7-4)}m(x)/g(x)$, los coeficientes de ese resto seran los bits de la paridad buscada. Operando se llega a que la paridad es 011 y el mensaje codificado queda 0111001.

2.3. Modulación OFDM

En 1966, Chang presenta un método para lograr la multiplexión de canales de datos a través de un medio de frecuencia acotada, en [3], que elimina los efectos de interferencia intersimbólica e intercanal. Implicó un cambio importante en la teoría de telecomunicaciones de la época, pues hasta entonces, los resultados existentes tomaban como funciones modulantes ortogonales, señales limitadas en el tiempo, lo que implica grandes anchos de banda en frecuencia, y en los canales de banda acotada implementados en la práctica se traducían en interferencias producto de los recortes en banda.

En el paper, Chang postula la idea de una nueva clase de funciones modulantes acotadas en frecuencia, que además, permite modular de manera independiente amplitud y frecuencia. En ese momento se sentaban las bases de la modulación OFDM (Orthogonal Frequency-Division Multiplexing).

Uno de los mayores problemas de los sistemas FDM, era la incapacidad para escalar en cantidad de canales. La complejidad y el costo de construir los osciladores para la cantidad de portadoras necesarias, mantenían la brecha entre la teoría y la práctica.

La solución a estos problemas, llega cuando se logra programar computadoras capaces de procesar grandes cantidades de datos, mediante la implementación del algoritmo de “Fast Fourier Transform”.

Weinstein y Ebert, resolvieron en [13] el problema de la escalabilidad de los sistemas FDM, mediante la conjugación de los mencionados avances tecnológicos, discretizando las señales a transmitir, y modulándolas por computadora, en lugar de usar los bancos de osciladores.

Una señal multitonal, puede ser vista como la transformada de Fourier de un tren de pulsos, y la demodulación coherente a su vez puede entenderse como la aplicación en tiempo continuo de una transformada inversa de Fourier. Entonces, probaron que muestreando la señal de origen, y mediante la implementación de un modem sobre una computadora que ejecute el algoritmo de la transformada rápida de Fourier, se pueden obtener aproximaciones suficientemente cercanas a los de la señal original.

En principio el resultado es válido para un sistema FDM con N canales simultáneos, con portadoras separadas en frecuencia en distancias suficientemente cercanas, como para aproximar la respuesta al impulso del canal, como si fuese de módulo constante sobre cada uno de los N canales.

Pero falta un paso más, pues la hipótesis del comportamiento del canal constante, se aleja bastante de la realidad. Plantearon entonces un canal, de respuesta al impulso lineal en frecuencia.

En estas condiciones, y atendiendo además que la señal discreta a transmitir solo “vive” en el ancho de banda de transmisión ($N \cdot \Delta f$), consideraron a la señal a transmitir como una transformada de Fourier enventanada. Desarrollando estas ideas lograron probar que, si la ventana es plana en las frecuencias de interés, y se agregan “guardas” de seguridad a ambos lados de las portadoras activas, de modo que las colas del enventanado caigan de forma continua, (alejando el enventanado

de la idealidad de las ventanas rectangulares) se logran condiciones para reducir la distorsión generada por la respuesta al impulso del canal, a efectos transitorios de dispersión rápida, y la señal en recepción sigue convergiendo a la señal transmitida.

2.4. MPEG y sus Estándares

El Moving Picture Experts Group (MPEG) [10] es un grupo de trabajo conformado por expertos internacionales, formado por la Organización Internacional de Normalización (ISO) en conjunto con la Comisión Electrotécnica Internacional (IEC), con el objetivo de desarrollar estándares para la codificación, compresión y transmisión de audio y video.

Uno de los estándares publicados por el MPEG, es MPEG-4. Consta de métodos para la compresión digital de contenidos audiovisuales, y abarca la difusión de los mismos a través de una amplia gama de tecnologías, desde el streaming de datos a través de la web, codificación de voz y video para telefonía y videoconferencias, comercialización de discos compactos (CD) y hasta formatos para la transmisión de Televisión.

Es en este último punto donde se vincula con ISDB-T Internacional, puesto que para la codificación de fuente en la norma, fue seleccionado el Estándar MPEG-4 Parte 10 “Advanced Video Coding”, también denominado H.264.

Para garantizar que los receptores de televisión digital ISDB-T, también sean compatibles con los transmisores tanto de ISDB-T como de ISDB-T Internacional, se encapsulan los videos codificados en H.264 dentro de un formato denominado “Transport Stream” que se define en la norma MPEG-2 Parte 1 – Sistemas.

En el transmisor gr-isdbt-tx, tomamos como fuente de datos un archivo codificado como Transport Stream, para garantizar esta compatibilidad.

2.4.1. MPEG 2 Transport Stream

Un Transport Stream (TS) es un contenedor de datos en el que se encapsulan en conjunto uno o más PES (Paquetized Elementary Streams), junto con códigos de corrección de errores y flags de sincronismo. La combinación de esta información, permite mantener la continuidad de la decodificación incluso cuando el canal se degrada fuertemente.

Un Paquetized Elementary Stream (PES), es una especificación de MPEG 2 para el transporte de flujos elementales, generalmente las salidas de los codificadores de audio y video. En ISDB-T, los Elementary Streams generalmente contienen video, audios en más de un idioma, archivos de subtítulo, grillas de programación y tablas de información de transmisión.

Al comienzo de la cadena de transmisión, en ISDB-T, se multiplexan varios TS, para crear un único TS sobre el cual se va a trabajar. El mismo será sometido luego a varias capas de codificaciones de canal, para robustecerlo aún más frente a las pérdidas. Este proceso se discutirá luego en el capítulo 3.

La estructura de un Transport Stream es la siguiente: (Imagen)

Capítulo 2. Fundamento Teórico

Los datos de los elementary streams se recortan en secciones de 188 bytes, este tamaño tan chico, permite que se realice un entrelazamiento con otros ES con muy baja latencia, y con una mayor resistencia ante las pérdidas.

2.4.2. Tablas PMT

Dentro de los Transport Streams se define el concepto de Programas. Cada programa esta definido en una tabla denominada PMT (Program Map Table), que viaja multiplexada en el TS de transmisión, identificada por un PID único. Los Elementary Streams asociados con el programa en cuestión, tienen sus PIDs listados en la PMT. En general, se asocia cada canal con un programa, aunque también podrían utilizarse programas para (completar)

Cuando un receptor decide reproducir un canal en particular, lo que tiene que hacer es decodificar los payloads contenidos en los TS cuyos PIDs están en la tabla PMT

Ademas de la tabla PMT, existen otros tipos de tablas en MPEG-2. Para el alcance de este documento, nos interesa detallar solo dos mas. La Program Association Table (PAT), contiene una lista con todos los programas

2.4.3. Tablas PAT

La Program Association Table (PAT), es una tabla que contiene una lista de todos los programas disponibles en el TS. Esta formada por valores de 16 bits denominados Program Number, asociados cada uno de ellos con un PID correspondiente a su tabla PMT dentro del stream.

De modo que, el receptor que se conecta al stream en cualquier momento de la transmisión, pueda hallar el programa de interés y sintonizarlo, es que se identifican a todos los paquetes del Transport Stream con un identificador único, o PID. Las tablas PAT siempre van contenidas en paquetes de PID 0x0000.

Habiendo identificado la PAT por su PID bien conocido, el receptor selecciona un programa de la lista y guarda el PID de su tabla PMT. Luego filtrando todos los paquetes cuyo identificador difiera del identificador del programa elegido, se hace con toda la información de decodificación del programa contenida en la PMT, y luego podrá discriminar entre todos los paquetes del TS, los del contenido audiovisual que le son de interés para decodificarlo.

2.4.4. Paquetes Nulos

Resulta vital para un esquema de transmisión de televisión mantener el bitrate constante, pero puede ocurrir que falten datos para contemplar esa tasa. Es por eso que se definen los paquetes nulos. Un multiplexor completa con los paquetes necesarios para mantener la tasa constante.

Para diferenciar los datos validos de los paquetes de relleno, al igual que lo que sucede con la tabla PAT, es que se usa para identificar los paquetes nulos un PID bien conocido, que en este caso es el 0x1FFF.

Capítulo 3

El Sistema de Televisión Digital Terrestre ISDB-T

El esquema de transmisión del estándar ISDB-T que se presenta en la Figura 3.2 puede ser caracterizado en cuatro grandes etapas que serán presentadas en este capítulo: el flujo de transporte BTS (*Broadcast Transport Stream*), la etapa de robustecimiento de la señal, la formación de los Cuadros OFDM con sus señales piloto y la puesta en el aire de la señal.

Cada una de esas etapas tiene a su vez varias subetapas, o bloques fundamentales, que se enfocan en resolver los distintos problemas discutidos que surgen al transmitir una señal inalámbrica. Esto da como resultado una variedad de parámetros de operación presentados en la Tabla 3.1 que permitirán por ejemplo, la transmisión jerárquica en distintas capas, cada una con su propia configuración; o enfocarse en robustecer la transmisión frente al *multipath* o al efecto Doppler.

A nivel de la distribución en el espectro, el estándar utiliza 6 MHz de ancho de banda de canal distribuido en 14 segmentos de los cuales sólo 13 son utilizados para enviar datos, el restante se utiliza como guarda a ambos lados del canal. También se incluye un piloto continuo que se trata de una portadora modulada en BPSK ubicada a continuación del Segmento 12.

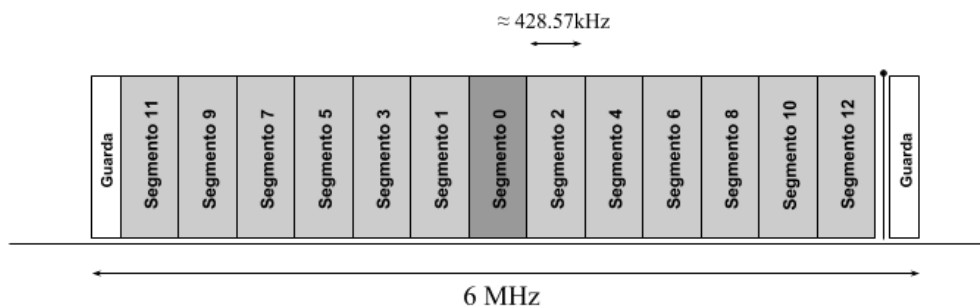


Figura 3.1: Distribución espectral de los segmentos OFDM. A la derecha, entre el segmento 12 y el intervalo de guarda, puede apreciarse la presencia del piloto continuo.

Capítulo 3. El Sistema de Televisión Digital Terrestre ISDB-T

Tener un esquema de transmisión jerárquica permite asignar a cada capa jerárquica la cantidad de segmentos que uno quiera. El estándar de la ARIB denomina a estas capas como A, B y C, y es posible tener configuraciones con solamente 2 capas presentes tal como la de la Figura 3.1 en la que la capa A corresponde al segmento 0 y el resto de los segmentos forman la capa B.

En los comienzos de la televisión digital en el Uruguay los operadores que prestaban servicios de televisión digital debían transmitir su señal en calidad HD, en SD y *oneseq*, es decir en tres capas jerárquicas. Esto era así porque se entendía que en el mercado circulaba una gran cantidad de receptores que no eran capaces de procesar la calidad HD. Posteriormente esa directiva fué suprimida y al día de hoy los operadores no están obligados a transmitir en SD.

El sistema puede operar en tres *modos de transmisión* diferentes que se caracterizan por la cantidad de portadoras utilizadas, siempre en el mismo ancho de banda de 6 MHz. Las portadoras utilizadas son $2^{10+modo}$ donde el modo puede ser 1, 2 o 3.

Se define la frecuencia de muestreo de la IFFT como $f_{IFFT} \triangleq \frac{512}{63\mu s} \approx 8,127MHz$, que a su vez coincide con el número de portadoras utilizadas sobre el tiempo de símbolo activo T_s . Por lo tanto la utilización de un número mayor de portadoras implica utilizar símbolos más largos para respetar la relación de la f_{IFFT} ; aumentar el tiempo de símbolo activo implica reducir el ancho de banda ocupado por cada portadora.

Cada segmento a su vez está conformado por dos tipos de señales portadoras. Las primeras son las correspondientes a los datos, y las otras son las denominadas *portadoras piloto* que cumplen se utilizan para la estimación del canal y transmisión de parámetros de control e información del sistema. De manera general, los segmentos tienen $96 \times 2^{modo-1}$ portadoras de datos $12 \times 2^{modo-1}$ portadoras piloto dependiendo del modo de transmisión. En la Tabla 3.1 se evalúan estos parámetros para los distintos modos de transmisión.

3.1. BTS como fuente de datos

El estándar ISDB-T admite la posibilidad de tomar hasta tres *Transport Streams* (TS) MPEG-4. A estos flujos se les debe agregar la información necesaria para la transmisión en capas jerárquicas. El bloque TS Remux es el que se encarga de multiplexar los tres flujos de transporte y a cada TSP agregarle 16 bytes de información. Una vez que se agregan estos datos los paquetes de cada capa son multiplexados según un patrón de ordenamiento que es único para cada configuración del sistema, en [7] se explica este patrón y se presenta un algoritmo para recuperarlo en recepción. El transmisor implementado en este trabajo toma como flujo de entrada un BTS ya conformado, y con la información jerárquica de los TSP es que logra procesar cada capa por separada. De ahí en adelante las capas son entrelazadas y moduladas cada una de acuerdo a su propia configuración.

3.1. BTS como fuente de datos

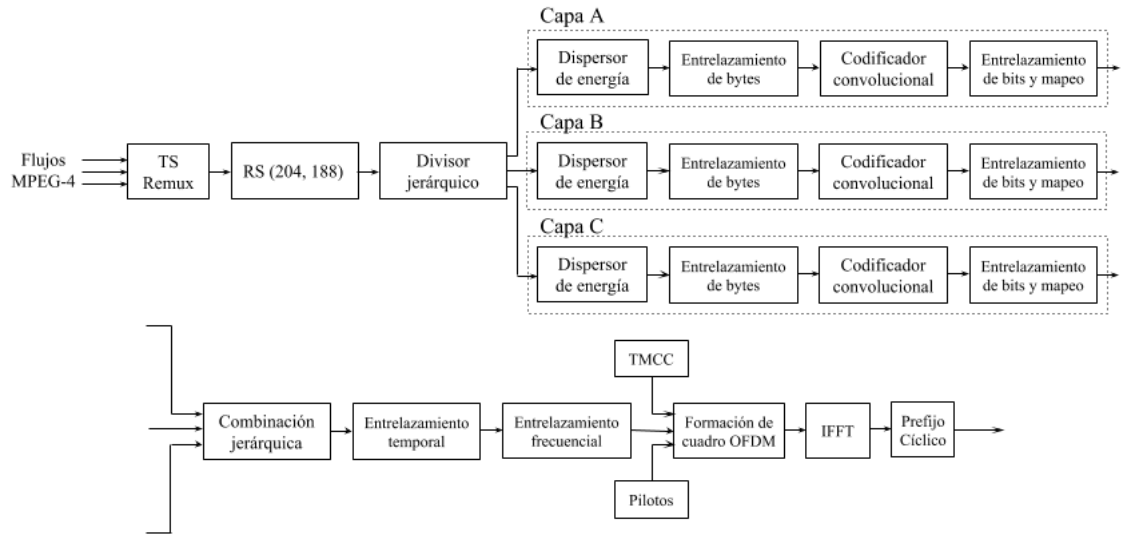


Figura 3.2: Diagrama de bloques del transmisor ISDB-T definido por la ARIB.

Parámetro	Valor
Ancho de banda del canal	6 MHz
Cantidad de segmentos	13
Ancho de banda de cada segmento	$6000/14 \approx 428,57 kHz$
Cantidad de portadoras activas por segmento	96 de datos y 12 pilotos (Modo 1) 192 de datos y 24 pilotos (Modo 2) 384 de datos y 48 pilotos (Modo 3)
Duración de símbolo activo	$252\mu s$ (Modo 1) $504\mu s$ (Modo 2) $1008\mu s$ (Modo 3)
Duración del prefijo cíclico	$1/4, 1/8, 1/16, 1/32$ (fracción del símbolo activo)
Tasa de código convolucional	$1/2, 2/3, 3/4, 5/6, 7/8$
Tasa de código Reed-Solomon	(188, 204)
Profundidad del entrelazamiento temporal	0, 1, 2, 4 (Modo 1) 0, 2, 4, 8 (Modo 2) 0, 4, 8, 16 (Modo 3)
Esquemas de modulación	DQPSK, QPSK, 16QAM, 64QAM
Frecuencia de muestreo (f_{FFT})	$512/63 \approx 8.127$ MHz

Tabla 3.1: Parámetros relevantes en el estándar ISDB-T.

3.2. Robustecimiento frente a las no idealidades del canal

Como primer medida para proteger los datos del BTS se aplica un código Reed-Solomon (204, 188). El proceso de codificación consiste en sustituir los últimos 16 bytes de los TSP por una paridad que permite corregir hasta 8 bytes de error en el paquete.

Este bloque remueve la información jerárquica agregada por el TS Remux con lo cual una vez que los TSP son codificados por el Reed-Solomon en principio ya no es posible distinguir a qué capa pertenece cada paquete, es decir que al llegar al bloque de división jerárquica ya no se cuenta con esa información. Una estrategia para sortear este problema, y de hecho la que se implementa en *gr-isdbt-tx*, consiste en separar primero los paquetes correspondientes a cada capa jerárquica y luego codificar los paquetes. Una vez encaminados los paquetes se puede prescindir de la información jerárquica. El diagrama de bloques presentado en la Figura 3.2 corresponde al esquema original de la ARIB en la que se define el estándar ISDB-T, mientras que la Figura 3.2 se encuentra la implementación utilizada en *gr-isdbt-tx*.

Para evitar los problemas de sincronismo que podría ocasionar una señal con muchos ceros o unos consecutivos, se utiliza un *dispersor de energía* que se encarga de generar un flujo pseudoaleatorio. Este proceso se define de manera tal que resulta ser totalmente invertible y por lo tanto aplicando el mismo proceso en recepción se obtiene la secuencia original.

Los canales inalámbricos son propensos a una multiplicidad de fuentes de error. Un tipo de errores muy comunes en estos canales son los *errores en ráfaga* provocando que una secuencia consecutiva de bits se corrompan. El impacto de estos tipos de errores crece con la velocidad de transmisión; a mayor velocidad de transmisión un mismo error afecta a una mayor cantidad de bits. Resulta necesario lograr algún tipo de inmunidad ante estos errores, es por ello que en muchos sistemas, y en particular en ISDB-T, se suelen concatenar códigos correctores de errores. David Forney en su trabajo *Concatenated Codes* [8] estudia el compromiso que existe en la utilización de sucesivos códigos concatenados, y en especial la performance que pueden llegar a alcanzar los códigos Reed-Solomon concatenados. El estándar ISDB-T utiliza este enfoque a través de la implementación de un código Reed-Solomon como *outer code* y un código convolucional como *inner code*. Como medida adicional para contrarrestar los efectos del canal se utiliza el *entrelazamiento de bits* que esencialmente consiste en introducir retardos variables en los bits que se transmiten.

Al transmitir en múltiples portadoras ortogonales se corre el riesgo de atravesar canales selectivos en frecuencia y por lo tanto que ciertas portadoras se vean continuamente afectadas. Para mitigar esto una estrategia puede ser rotar de portadora los contenidos que se transmiten, es decir hacer un *entrelazamiento frecuencial*. Con esto se logra que ante un canal selectivo, ya lo sea por sus características propias o por interferencias de señales espurias fuera de banda, las portadoras que son afectadas no resulte en una pérdida de bloques de información irrecuperables.

3.3. Formacion de los cuadros OFDM

El estándar define una estructura de cuadro OFDM formada por un conjunto de 204 símbolos OFDM consecutivos, es decir que, por ejemplo, para el modo 1 se tienen 108 símbolos complejos por segmento con lo cual un cuadro OFDM en modo 1 está compuesto por $108 \times 13 \times 204$ complejos. Esta estructura facilita las tareas de sincronización en el receptor.

3.4. Las portadoras y la modulación

Cada segmento OFDM está conformado por una cantidad de portadoras en función del modo de transmisión. La mayor cantidad de portadoras está destinada a la transmisión de los datos mientras que algunas otras son utilizadas como mecanismos para la estimación del canal, sincronización, transmisión de parámetros de funcionamiento entre otros.

Scattered Pilot Las portadoras piloto son señales moduladas en BPSK que se generan a partir de una secuencia pseudoaleatoria PRBS (Pseudo Random Binary Sequence). El procedimiento para generar estas portadoras consiste en evolucionar el circuito de la Figura XX y tomar W_i para la modulación BPSK. El índice de la portadora SP dentro del símbolo OFDM es el que determina la cantidad de veces que se debe evolucionar el registro para obtener W_i . Por último la modulación se realiza como $(4/3, 0)$ si $W_i = 0$, y $(-4/3, 0)$ si $W_i = 1$.

Las SP se ubican cada 12 portadoras y a medida que transcurren los símbolos OFDM las mismas van rotando $3 \times n \text{ MOD } 4$, donde n es el número de símbolo OFDM. Esto sucede así puesto que la función de estas portadoras es justamente la estimación del canal y por lo tanto para contemplar la situación de todo el canal de 6MHz es necesario ir rotándolas.

Portadora TMCC Las portadoras TMCC son las que transportan toda la información de las capas jerárquicas y la información de control. Si bien en los cuadros OFDM se transmiten varias TMCC estas tienen exactamente el mismo contenido, lo que da al receptor la posibilidad de ponderar todas las TMCC que recibe para quedarse con las que tienen mayor porcentaje de coincidencia entre sí. Se compone de 204 bits modulados en DBPSK de los cuales el primero es un bit de referencia para realizar la modulación diferencial. Este bit coincide con el valor de la W_i correspondiente al primer símbolo del cuadro OFDM. Los siguientes 16 bits son utilizados por una señal de sincronismo que puede tomar los valores $w_0 = 0011010111101110$ y $w_1 = 1100101000010001$, alternándose entre una y otra en cada cuadro OFDM. A continuación se encuentran los datos de la información jerárquica y otros parámetros para los que se dedican 105 bits. Estos 105 bits son protegidos por un código cíclico acortado (200,118) con lo cual se dedican los siguientes 82 bits para la paridad.

Portadoras Auxiliares

Las portadoras auxiliares están pensadas para transmitir información complementaria del control de transmisión. Utilizan modulación diferencial DBPSK con referencia W_i al igual que la TMCC. Su uso es opcional y el sistema puede

Capítulo 3. El Sistema de Televisión Digital Terrestre ISDB-T

trabajar perfectamente prescindiendo de ellas; en caso de no ser usadas los bits correspondientes a las portadoras AC se rellenan con "1".

Piloto Continuo Al igual que las SP, se agrega un piloto continuo a la derecha del segmento 12 con propósitos de sincronismo en recepción. Su modulación también es BPSK y toma el valor de $(-4/3, 0)$, $(4/3, 0)$, $(4/3, 0)$ para los modos 1, 2 y 3 respectivamente.

3.5. La puesta en el aire de la señal

Luego de conformarse el cuadro OFDM los datos ya están listos para ser transmitidos mediante la modulación OFDM. En este punto los símbolos son procesados por el bloque de la IFFT y a continuación es necesario agregar el prefijo cíclico. Una vez agregado el prefijo las muestras ya están prontas para ser dirigidas al conversor D/A para la transmisión en ondas EM.

3.5.1. Necesidad del prefijo cíclico

Cuando se pasa de trabajar en el dominio continuo al dominio discreto muchas cosas no funcionan igual. Si bien es cierto que en el dominio continuo la transformada de Fourier de un producto convolución coincide con la transformada de cada factor, esto deja de ser válido en tiempo discreto; si $y[n] = x[n] * h[n] \Rightarrow DFT\{y[n]\} \neq X[q]H[q]$. Sería bueno poder tener una relación similar a la que ocurre en tiempo continuo y así poder separar fácilmente las muestras que se enviaron de la respuesta del canal.

Afortunadamente existe un tipo de producto convolución especial llamado *convolución circular* que nos devuelve esa propiedad de la transformada de Fourier. Supongamos que la señal que enviamos es $x[n]$ y la respuesta del canal $h[n]$ con $n = 0, \dots, N-1$. La convolución circular de dos señales $x[n]$ y $h[n]$ se define de la siguiente manera:

$$x[n] \circledast h[n] = \sum_{k=0}^{N-1} x[(n-k) \text{ MOD } N] h[k] \quad (3.1)$$

Para evitar sobrecargar la notación se suele utilizar la siguiente nomenclatura $x_N[n-k] \triangleq x[(n-k) \text{ MOD } N]$. Con lo cual la convolución circular queda expresada:

$$x[n] \circledast h[n] = \sum_{k=0}^{N-1} x_N[n-k] h[k] \quad (3.2)$$

La señal $x_N[n-k]$ no es otra cosa que la señal $x[n-k]$ periodizada cada N muestras. Tomemos la transformada discreta de Fourier de esta convolución circular y veamos lo que sucede.

$$Y[q] \triangleq DFT\{x[n] \circledast h[n]\} = \sum_{n=0}^{N-1} e^{-j2\pi \frac{nq}{N}} \sum_{k=0}^{N-1} x_N[n-k] h[k] \quad (3.3)$$

3.5. La puesta en el aire de la señal

$$Y[q] = \sum_{k=0}^{N-1} h[k] \sum_{n=0}^{N-1} e^{-j2\pi \frac{nq}{N}} x_N[n-k] \quad (3.4)$$

Renombrando la variable m como $n - k$ se tiene:

$$Y[q] = \sum_{k=0}^{N-1} h[k] \sum_{m=-k}^{N-1-k} e^{-j2\pi(m+k)\frac{q}{N}} x_N[m] \quad (3.5)$$

Todo lo que no depende de m en la segunda sumatoria puede salir como factor

$$Y[q] = \sum_{k=0}^{N-1} e^{-j2\pi \frac{mq}{N}} h[k] \sum_{m=-k}^{N-1-k} e^{-j2\pi \frac{mq}{N}} x_N[m] \quad (3.6)$$

En la ecuación 3.6 puede apreciarse que la sumatoria en k corresponde a la $DFT \{h[k]\}$. La otra sumatoria resulta un poco más complicado de verla pero pensemos que $x_N[m]$ es la señal $x[m]$ de N muestras periodizada; la sumatoria se lleva a cabo en una ventana de N muestras por lo cual esto coincide con la transformada discreta de Fourier de la señal $x[n]$. Dicho de otra manera, la sumatoria de la ecuación 3.6 es una manera de calcular la DFT en la que se suman los términos en otro orden, pero al final la suma es la misma.

Con esto queda demostrado que la DFT de la convolución circular de dos señales coincide con el producto de sus transformadas discretas, es decir que $DFT \{h[n] \otimes x[n]\} = H[q]X[q]$.

Pasemos a ver qué sucede con la señal OFDM cuando se propaga por el canal inalámbrico. En principio el canal podría variar con el tiempo, con lo cual el modelo discreto de la señal recibida queda de la siguiente manera:

$$r[n] = \sum_{k=-\infty}^{\infty} h_k[n] x[n-k] + w[n] \quad (3.7)$$

Donde $w[n]$ corresponde al ruido gaussiano aditivo que introduce el canal, y $h_k[n]$ es la respuesta al impulso del canal que en principio puede ser una función del tiempo. Para el estudio que realizaremos se asumirá que el canal es un sistema lineal invariante en el tiempo y su respuesta está definida por L muestras, con lo cual se elimina la dependencia de n :

$$r[n] = \sum_{k=0}^{L-1} h_k x[n-k] + w[n] \quad (3.8)$$

La señal resultante tendrá $N + L - 1$ dado el efecto que tienen las muestras de la respuesta al impulso del canal. Esto hace que al enviar dos símbolos OFDM consecutivos haya un solapamiento de $L - 1$ muestras en recepción. Como primera medida podría separarse los símbolos enviados la cantidad de muestras necesarias, sin embargo existe otra solución que consiste en el agregado de un *prefijo cíclico*. La idea es utilizar esos L espacios para transmitir las últimas L muestras del símbolo siguiente tal como se muestra en la Figura 3.3.

Capítulo 3. El Sistema de Televisión Digital Terrestre ISDB-T

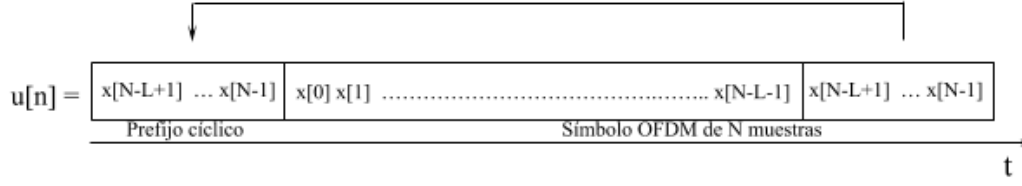


Figura 3.3: Inserción del prefijo cíclico.

El estándar ISDB-T ofrece la posibilidad de utilizar distintos largos de prefijo cíclico. Una manera de establecer sincronismo, y también poder conocer los parámetros básicos de transmisión desde el lado del receptor como el modo y el largo del prefijo cíclico, consiste en tomar símbolos OFDM consecutivos que van arribando y retardarlos una cierta cantidad de símbolos. Luego se calcula una función de verosimilitud propuesta por Van de Beek en [12] y según su característica es que se puede determinar el largo del prefijo cíclico y el modo de transmisión.

La nueva señal conformada a partir de $x[n]$ con su prefijo cíclico queda expresada de la siguiente manera:

$$u[m] = \begin{cases} x[N - L + m + 1] & \text{si } m = 0, \dots, L - 2 \\ x[m - L + 1] & \text{para } m = L - 1, \dots, N + L - 2 \end{cases} \quad (3.9)$$

Ahora con esta nueva señal que hemos definido, en recepción llegará la siguiente señal:

$$v[n] = \sum_{k=0}^{L-1} h[k]u[n - k] + w[n] = \sum_{k=0}^{L-1} h[k]x[n - L - k] + w[n] \quad (3.10)$$

La superposición de las primeras $L - 1$ de v hace que deban descartarse. Por lo tanto se considerarán las muestras con $n = L, \dots, L + N - 1$ y entonces es posible considerar un n' el cual contempla que las muestras indicadas ya han sido descartadas. Como este es un proceso cíclico símbolo a símbolo, la ecuación 3.10 queda así:

$$v[n'] = \sum_{k=0}^{N-1} h[k]x_N[n' - k] + w[n'] \text{ con } n' = [0, \dots, N - 1] \quad (3.11)$$

Esta última ecuación no es otra cosa que la convolución circular $h \circledast x$ y que, como se vió anteriormente, su transformada de Fourier resulta ser $Y[q] = H[q]X[q]$. De esta manera el receptor una vez que realizó su estimación del canal es capaz de separar las muestras $X[q]$ y así recuperar la señal transmitida.

3.5.2. La potencia requerida para transmitir

- Relación entre la potencia necesaria segun la modulación - SNR Necesario para decodificar

Capítulo 4

Radio definida por Software

Cuando en telecomunicaciones se habla de radio, se hace referencia al hardware necesario para transmitir o recibir información, haciendo uso de la banda del espectro radioeléctrico ubicado en radiofrecuencia (RF).

La mayoría de los sistemas de comunicación de hoy en día necesitan de un radio, la telefonía móvil, la navegación por satélite, la aviación comercial, la televisión abierta, son todos ejemplos. En general los radios varían en dimensiones y costos según la aplicación, pero son muy poco versátiles en cuanto a sus capacidades. Con cada actualización tecnológica que reciben las redes, por ejemplo la modernización de la red móvil de 3G a LTE, implica una renovación total del equipamiento de radio.

El concepto de Radio Definida por Software no es nuevo, algunas de sus ideas fundacionales se publicaron en la década de los 70. Pero ha sido el masivo desarrollo de la industria tecnológica el que ha impulsado en los últimos años la implementación de sistemas basados en SDR.

Para implementar un equipo de radio definida por software, solo es necesario contar con una antena de radiofrecuencia, capaz de transmitir/emitir señales analógicas, y un conversor analógico-digital que alimenta de muestras a un procesador de propósito general. Es aquí donde radica la diferencia con los radios implementados en hardware, en los cuales se implementa toda una circuitería electrónica para realizar el procesamiento. La masividad de los procesadores y el auge de la programación de hoy en día, permiten implementar fácilmente un SDR, y realizar todo tipo de proyectos de distinta escala, reutilizando el equipamiento.

Durante la realización de este proyecto, utilizamos como entorno de desarrollo para crear los bloques de procesamiento de señales al software GNU Radio, y como antena utilizamos un equipo N200 del fabricante Ettus Research.

4.1. GNU Radio

GNU Radio es un entorno de desarrollo orientado a procesamiento de señales, es gratuito y de código abierto. Mediante la interconexion de bloques de procesamiento, puede usarse en conjunto con antenas de RF para desarrollar SDRs, o en

Capítulo 4. Radio definida por Software

una versión local en modo de simulación de sistemas.

La aplicación por defecto trae una amplia gama de bloques funcionales para trabajar, desde herramientas simples como filtros y ecualizadores, hasta estructuras complicadas, como lo es un transmisor DTV completo. Existe además una gran comunidad, muy activa, que está permanentemente publicando nuevos contenidos, para ampliar la gama de herramientas existentes en la actualidad.

Esto se da porque al ser de código abierto, es relativamente sencillo crear bloques nuevos. El entorno soporta desarrollos de código en python y en C++, siendo este último el lenguaje con el que hemos implementado nuestro transmisor `gr-isdbt-tx`. Si bien puede parecer un poco complejo comprender la metodología de GNU Radio en un principio, una vez que comenzamos a trabajar con él y atravesamos la curva de aprendizaje, nos encontramos con una herramienta muy poderosa, y con la que es más sencillo comprender algunos de los conceptos teóricos establecidos en la norma. Durante la carrera, nos habíamos encontrado con GNU Radio en algunas ocasiones, donde pusimos en práctica algunos conceptos básicos de sistemas de comunicación. No fue hasta esta instancia, en la que pudimos encontrarnos con el potencial total de la herramienta.

Pasaremos a desarrollar algunos de los conceptos clave para comprender el uso de GNU Radio, ejemplificando con nuestro transmisor `gr-isdbt-tx`.

4.1.1. Flowgraphs

El flowgraph es la estructura de datos más básica del programa. Puede entenderse al mismo como si fuera una mesa de trabajo. Dentro de un flowgraph, vamos a insertar nuestras muestras desde la fuente, las vamos a procesar y las vamos a exportar, ya sea hacia un archivo o hacia el canal a través de una antena.

4.1.2. Bloques

En GNU Radio, los datos siempre se mueven a través de bloques. Un bloque puede tener la fuente de datos, realizar operaciones sobre las muestras, puede exportar datos hacia el exterior del entorno, o puede tener toda una estructura de bloques adentro. Existen cuatro tipos de bloques, y se diferencian por la tasa de muestras que atraviesan el mismo.

Bloques Sincronos (1:1), este bloque permite implementar funcionalidades que consuman y produzcan la misma cantidad de muestras por puerto. Los bloques sincronicos pueden tener cualquier cantidad de entradas y salidas. En particular, cuando un bloque sincrónico tiene 0 entradas, se tiene una fuente de datos (source). Cuando, al contrario, tiene 0 salidas, decimos que es un sumidero (sink). El bloque dispersor de energía por ejemplo, es un caso de bloque sincrónico de `gr-isdbt-tx`. Volveremos sobre esto en el capítulo 5.

Bloques Decimadores (N:1), son los que producen menos muestras de las que consumen, y de forma análoga, existen también los *Bloques Interpoladores (1:M)*.

Finalmente tenemos el caso general, *Bloques Generales (N:M)*, en el cual no existe a priori una relación entre muestras de entrada y de salida, y cada usuario

debe definirlo para el bloque en particular que esta creando mediante una función llamada `forecast`. Uno de los mas claros casos de bloque general en nuestro transmisor, es el bloque que crea el cuadro OFDM. En tiempo de compilación, no sabemos los parámetros de transmisión que determinan la tasa de muestras. Es en tiempo de ejecución que estos valores quedan definidos, el propio motor de GNU Radio se encarga de orquestar el flujo de datos para que todo funcione en régimen, Para implementar el transmisor, tuvimos que separar las funcionalidades en bloques de procesamiento, e implementar aquellos bloques que no estuviesen ya presentes en el repositorio. Algunos ejemplos como conversores serie a paralelo, el codificador Reed Solomon, el codificador Viterbi, existían ya como bloques funcionales del programa, y fueron re utilizados.

4.2. Hardware

Universal Software Radio Peripheral (USRP) es una linea de plataformas de hardware para SDRs diseñados y comercializados por la empresa Ettus Research. Gran parte de la arquitectura de estos equipos es de código abierto, y puede ser descargada desde la web de la empresa. Con estos equipos se logra poner en funcionamiento soluciones de radio definida por software por costo accesible.

En líneas generales, se componen de una placa madre en la que conviven varios subsistemas, como lo son un generador y sincronizador de señales de reloj, conversores analógico digitales y FPGAs, que se encargan de realizar tareas de procesamiento de señal en banda base. Se complementa con una placa modular denominada “daughterband” que realiza tareas como filtrado, condicionamiento de señal y modulación las señales en frecuencias de hasta 6 GHz.

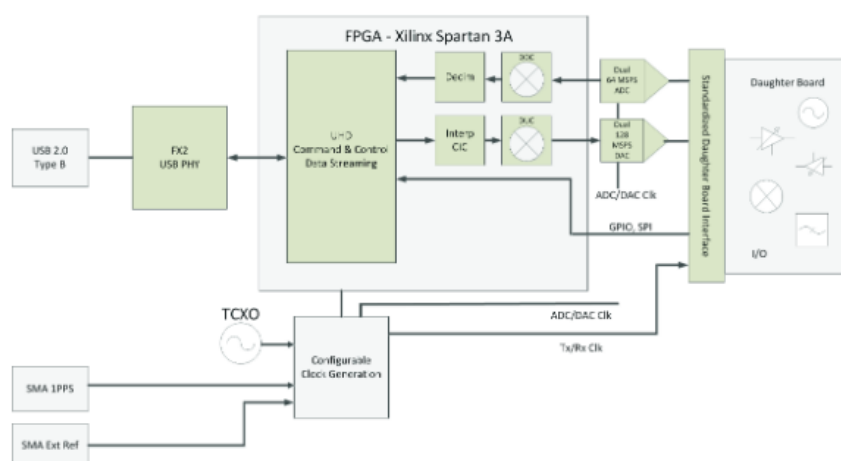


Figura 4.1: Arquitectura general de un equipo USRP

Durante este proyecto, trabajamos con el modelo USRP B100.

Capítulo 4. Radio definida por Software

4.2.1. USRP B100

En este modelo, las muestras llegan al equipo a través de una interfaz USB 2.0 y tiene una capacidad para transmitir en velocidades de hasta 16 MS/s, cuando se trabaja con muestras de 8 bits.

Este modelo en particular está actualmente discontinuado, pero su precio en el mercado rondaba los 700 dólares. Para el desarrollo del proyecto, recibimos en calidad de préstamo uno de estos equipos para las pruebas.

Para poder levantar el equipo desde una PC, es necesario instalar el driver universal de Ettus, “USRP Hardware Driver”. La mejor forma de hacer esto es instalarlo desde su repositorio oficial [11]. El driver es gratuito y de código abierto, y soporta todos los equipos de radio de Ettus.

La forma más sencilla de instalar el driver, es ejecutar en una línea de comandos `sudo apt-get install uhd`. Pero este comando instalara por defecto, el último release estable del controlador. Puede suceder que se desee trabajar con un release en particular, para eso, se debe realizar una copia del repositorio con las herramientas de línea de comando de *git*.

```
git clone https://github.com/EttusResearch/uhd
```

Nos creará una copia del repositorio entero del driver. Una vez ahí, es necesario moverse hacia el commit donde se encuentra el release que nos interesa instalar, para eso, nos movemos entre las branches del repositorio con el comando:

```
git tag -l
```

Obtenemos una lista de los releases publicados en el repositorio, para movernos al de interés, utilizamos el comando

```
git checkout nombre_release_deseado
```

Una vez parados en el release a instalar, simplemente instalamos con el uso de la herramienta *make*

Con el driver instalado, para verificar que esté funcionando correctamente, debemos realizar los siguientes pasos.

```
uhd_usrp_probe
```

Muestra una lista de los dispositivos soportados que estén conectados a la pc. En nuestro caso, el equipo es el B100 y la salida del comando es la siguiente:

También se puede configurar antenas que no estén directamente conectadas a la pc, pero sí en una dirección de red. No vamos a entrar en más detalles con respecto a este punto, pero vale la pena mencionarlo.

Interfaz con GNU Radio

Para intercomunicar al equipo con el software de procesamiento de datos, es necesario incluir en el flowgraph, bloques del complemento gr-uhd (en las versiones actuales de GNURadio ya vienen incluidos). Este complemento, vincula a la interfaz con el driver que se instaló según los detalles en la sección anterior. En gr-isdbt-tx, las muestras ya procesadas se enviaban al equipo a través de un bloque sink.

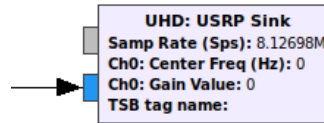


Figura 4.2: Bloque sink al equipo USRP

Los argumentos con los que funciona son la tasa de muestras, la frecuencia de destino en la que se desea transmitir y el valor de ganancia aplicado a la señal de entrada. Es importante verificar que los parámetros a seleccionar sean soportados por el hardware. En el caso de nuestra transmisión, tiene una tasa de muestras objetivo de:

$$f_{IFFT} = \frac{\text{portadoras}}{T_s} = \frac{512}{63} \approx 8,127 \text{ MHz}$$

Para verificar que la tasa de muestras sea soportada por el transmisor, primero debemos verificar el tamaño de los datos. En nuestro caso, el tipo es *grc::complex*, el cual es un tipo propietario de GNU Radio, formado por dos variables *std::complex* cada una de 32 bits representadas en punto flotante. Es decir que cada una de las muestras de nuestro transmisor se compone de 64 bits.

Recordando que el B100 tiene una capacidad de $61,44 \text{ MS} \bar{s} @ 16 \text{ bit} I/Q$, si le imponemos nuestra tasa de $32 \text{ bit} I/Q$ tendremos un ancho de banda disponible aproximado de $\approx 32 \text{ MS} \bar{s} @ 32 \text{ bit}$ lo cual es muy superior a nuestra tasa objetivo. Esto nos garantiza que podremos transmitir sin problemas de aliasing.

Otro aspecto a tener en cuenta con el transmisor, es verificar que la amplitud de las muestras esté normalizada, a modo de evitar un efecto de *clipping* a la salida, esto es, que todas las muestras superiores a 1, terminaran con la misma amplitud en recepción, lo que nos dejara una señal trunca en amplitud. Evitar este problema resulta sencillo, basta con agregar una ganancia a la salida del sistema de factor 0.5, para que las amplitudes que llegan al bloque de sink, y van hacia el equipo estén lejos de los umbrales de clipping.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 5

Un transmisor ISDBT implementado en GNU Radio

5.1. El flujo de datos en GNU Radio

Para el desarrollo de los bloques de `gr-isdbt-tx`, fue necesario comprender el modo en el que los datos se mueven entre bloques en GNU Radio. El motor del programa realiza llamados de forma periódica a los bloques en el `flowgraph`, dependiendo de la cantidad de muestras que tenga en cola para procesar en cada bloque, y de la tasa de muestras que busque mantener constante a través del sistema, le comunica a cada bloque, la cantidad de datos que necesita que le dejen en salida.

Al momento de crear un nuevo bloque, debemos especificar en la firma de la función, los parámetros necesarios para que el motor central pueda controlar de flujo de datos de nuestro nuevo bloque. Para realizar esto, se utilizan ciertos parámetros, precargados en la estructura de clases de los bloques, orientados a la entrada/salida de datos, estos son, `noutputitems` y `ninputitems`.

Mediante estas variables, el motor de procesamiento puede obligar a los bloques a ejecutarse mas de una vez en cada llamado, pues exige la cantidad de salidas que necesita para mantener la tasa. Para esto, le notifica al bloque en la variable `noutputitems`, la cantidad de “tandas” de datos que la instancia actual del bloque necesita devolverle al motor de procesamiento para mantener la tasa.

La variable `ninputitems`, mantiene el control de la cantidad de datos que hay en cada uno de los puertos de entrada. Y utilizándola como variable, podemos especificar cuantos datos de entrada precisamos consumir para obtener un dato en el puerto de salida.

Esta operación depende bastante de la naturaleza del bloque, por ejemplo, en los bloques síncronos, se sabe de antemano que la tasa de muestras se mantendrá constante, por lo que alcanza con especificar el tipo y la cantidad de muestras que atravesaran el bloque en una ejecución.

Los bloques de tipo general, son mas complicados, pues es necesario especificar explícitamente la relación entre la cantidad de muestras a la entrada y a la salida. Este control de flujo se realiza desde un método, precargado en la clase con la que

se crea el bloque, denominado forecast.

5.2. Obtencion de los TSP por capa

Como se explico anteriormente, en un solo flujo de transporte coexisten paquetes pertenecientes a las distintas capas de transmisión, sino que también están las tablas de información, y los paquetes nulos para mantener el flujo de datos constante.

Es necesario separar al principio del transmisor, esta información, para procesarla individualmente cada capa, afín de lograr mantener los programas contenidos en la misma en un mismo flujo, pues como vimos, las capas pueden tener distintas modulaciones y distintos retardos.

Cada paquete tiene en su encabezado IIP, un parametro denominado *layer_information*, que contiene información sobre la capa de transporte a la que pertenece. El parametro ocupa los primeros cuatro bits del segundo byte del encabezado del TSP, y se decodifica según la siguiente tabla

Valor $b_7b_6b_5b_4$	Capa
0000	Null TSP
0001	Capa A
0010	Capa B
0011	Capa C

Tabla 5.1: Parámetro indicador de capa en el IIP.

El primero de los bloques de gr-isdbt-tx, es un bloque que se encarga de realizar esta tarea. Dado que conocemos de antemano la cantidad de TSPs que estan contendios en un cuadro multiplex de nuestro BTS de prueba, definimos un bloque con un puerto de entrada y tres puertos de salida. En cada ejecucion, se recorre un cuadro multiplex, analizando para cada uno de los TSP el parámetro *layer_information*, decodificamos a que capa pertenece, y se enrutan según corresponda.

Entendemos que la implementación que presentamos de este bloque, existen cosas a mejorar. No las incluimos en este release del código por cuestiones de tiempo. De primera mano, la cantidad de TSP por cuadro multiplex podría ser calculada en tiempo de ejecución, para el BTS que el usuario quiera usar en ese momento, al igual que la cantidad de paquetes de cada capa. Hoy esos parámetros están fijados, y eso limita la robustez del transmisor. Discutiremos estos temas mas a fondo en el capitulo 7.

5.3. Codificaciones de Canal

5.3.1. Reed Solomon

La norma ISDBT utiliza una implementación exactamente similar a la de DVB-T del codificador Reed Solomon. En cada TSP se sustituyen los últimos 16 bytes del IIP por la palabra de redundancia de un código RS(204,188). Para lograr esta distancia, lo que se hace es llevar el payload hacia un tamaño más grande, agregando 51 bytes en 0 al comienzo del TSP. Luego, el nuevo payload de 239 bytes, se inyecta en un codificador RS(255,239).

El cambio de tamaño se realiza principalmente para utilizar un algoritmo más eficiente. Una vez obtenida la palabra código del mismo, se remueven los primeros 51 bytes nulos, y obtenemos el código RS(204,188) que entra en el tamaño de un TSP, y además es capaz de corregir errores hasta de 8 bytes.

Para la implementación de esta funcionalidad en ISDB-T, no fue necesario desarrollar ningún bloque. Dentro de los complementos de GNU Radio está el paquete gr-dvb [2] (originalmente un proyecto de terceros, que en 2015 fue incorporado al repositorio oficial de GNU Radio), que contiene los bloques necesarios para implementar tanto un transmisor como un receptor bajo la norma DVB. Al compartir el codificador Reed Solomon con ISDBT, bastó con utilizar el bloque de gr-dvb.

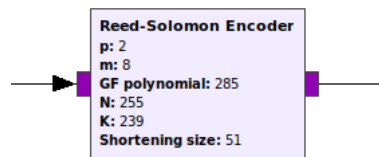


Figura 5.1: Bloque Reed Solomon Encoder de gr-dvb.

En la figura 5.1, mostramos el bloque y la configuración paramétrica del mismo, tal cual se implementó en gr-isdbt-tx. Los parámetros **p**, **m** y **GF polynomial** son los que configuran el polinomio generador que es la base del algoritmo. Para lograr el polinomio generador del algoritmo para RS(255,239), debemos utilizar el polinomio:

$$p(x) = x^8 + x^4 + x^3 + x^2 + 1$$

5.3.2. Viterbi

Otro de los códigos de canal implementados por ISDBT es el código Viterbi. Este código, es convolucional con puncturing, su código madre tiene una tasa de $\frac{1}{2}$ y constante $k = 7$, lo que permite obtener en salida datos codificados a cualquier tasa m/n en transmisión, sin aumentar la complejidad de la decodificación en recepción.

Capítulo 5. Un transmisor ISDBT implementado en GNU Radio

Esto sucede, porque tanto el lado transmisor como el receptor, conocen la llamada matriz de puncturing. En esa matriz, se especifica para cada tasa de código buscadas, los bits redundantes que serán eliminados al transmitir, para lograr la tasa deseada. Vale recordar que estos bits deben ser reingresados por el decodificador en recepción, pues de lo contrario aumentaría fuertemente la complejidad de la decodificación.

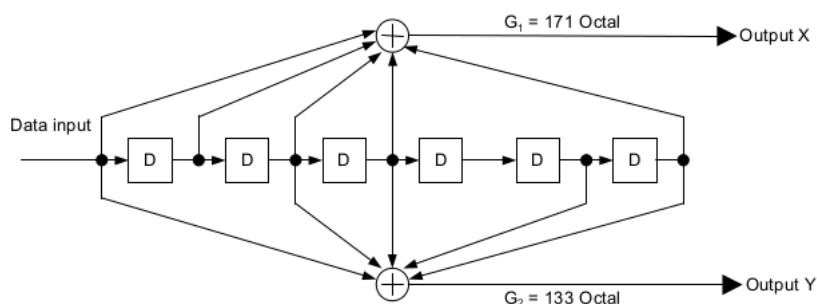


Figura 5.2: Circuito del código madre del Viterbi que implementa ISDBT.

Agregar este tipo de códigos a la cadena de transmisión, aumenta la resistencia ante las pérdidas y mantiene una tasa de bits constante, lo que colabora con el mantenimiento del sincronismo del sistema.

DVB utiliza el mismo código en su cadena de transmisión, con los mismos parámetros, por lo que en `gr-isdbt-tx`, no fue necesario realizar un desarrollo del bloque sino que reutilizamos el existente en `gr-dvb`.

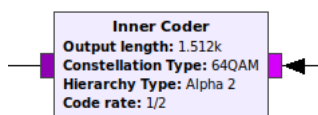


Figura 5.3: Bloque de `gr-dvb` que implementa el codificador Viterbi.

5.4. La modulación

Para resolver la modulación en nuestro transmisor, decidimos crear un solo bloque que resuelva en conjunto los problemas de interleaving de bits y modulación. Esto resultó bastante conveniente, pues bastó con crear una variable que contenga un selector de modulación, y en base a él se crean una serie de colas, cuyos tamaños responden a lo explicado sobre interleaving en el inciso 5.7.

El problema que nos encontramos en esta etapa, fue el de particionar la información en un byte, pero conservar el resto. Un ejemplo claro de esto se da cuando la modulación es *64-QAM*, pues cada símbolo en este caso se construye con 6 bits. Ahora, el parseo de datos entre bloques, GNU Radio lo hace encapsulados en datos de tipo *byte*. Por lo tanto, necesitábamos una forma de resolver el problema de

5.4. La modulacion

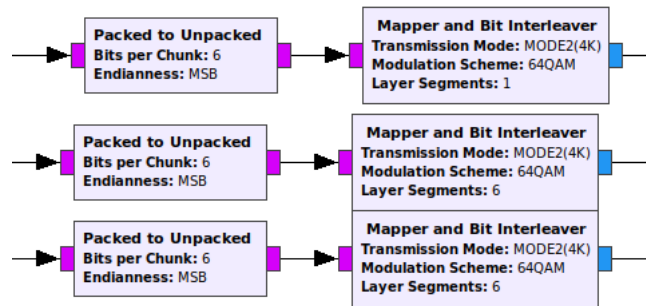


Figura 5.4: Implementacion de la modulacion e interleaving a nivel de bits en gr-isdbt-tx.

separar 6 bits utiles de un dato, y guardar los 2 siguientes para el proximo simbolo. En principio, no parecia ser un problema tan grave, pero podria pasar que los 2 bits que nos quedan pendientes, se correspondan con un simbolo que se construira en una proxima llamada al metodo modulador del objeto "bloque mapper". A nivel de programacion, esto implica que la memoria volatil de la instancia del objeto, se borra, por lo que habria que utilizar algun almacenamiento que permanezca en memoria estatica, y actualizarla en cada simbolo que se procesa.

Recorrer ese camino nos pareció ademas de muy laborioso, poco eficiente en terminos de CPU. Encontramos la solucion en un conjunto de bloques propietarios de GNU Radio, que lo resuelven de manera eficiente. Los bloques *Packed to Unpacked* nos resuelven el problema. Basta con tener en cuenta a la hora de utilizar el transmisor, definirle el parametro de *Bits per Chunk* de forma consistente con la modulacion que se este utilizando.

Modulación	Bits por símbolo
QPSK	2
16-QAM	4
64-QAM	6

Tabla 5.2: Bits por símbolo según modulación.

A la salida de los bloques *Bits per Chunk*, tendremos un byte completo de 8 bits, pero rellenado con la cantidad de bits requerida por la modulación, comenzando por el bit mas significativo. Esto nos permite normalizar la recepci3n de datos del bloque modulador, para cada caso, sabremos perfectamente como esta compuesto el byte de datos.

Una vez allí, alcanza con crear una cola con el retardo correspondiente, y enrutar los bits hacia la misma de forma ordenada. La secciona de modulación del bloque, funciona de la siguiente manera.

Se obtiene de la cola los bits a procesar, y se arma, manteniendo el orden especificado en la norma, una palabra de N bits, donde $N = 2, 4, 6$ según corresponda en la tabla TablaBits por simbolo segun modulacion. Una vez formada la palabra, se re interpreta como un numero decimal, y se busca en un arreglo el complejo que

se corresponda con la palabra recién formada. Esta información se obtiene de la tabla

5.5. El uso de los entrelazamientos

5.5.1. Entrelazamiento frecuencial

El entrelazamiento frecuencial consiste en permutar las portadoras de un símbolo OFDM en el dominio de la frecuencia. Dada la característica de ISDB-T de utilizar un gran número de portadoras, ocurre que frente a canales selectivos en frecuencia se pueden ver afectadas una cantidad importante de portadoras consecutivas, con lo cual la capacidad de corrección de los códigos resultaría superada.

Realizando un entrelazamiento frecuencial las portadoras que pudieran verse afectadas en un canal selectivo son desentrelazadas en recepción y, por lo tanto, los errores que pudieran haber quedan distribuidos facilitando la tarea de los códigos correctores.

El procedimiento del entrelazamiento en ISDB-T se divide en dos etapas: el *entrelazamiento inter-segmentos*, y el *entrelazamiento intra-segmentos*.

Primero se separan los segmentos en tres grupos: los destinados a recepción parcial (*one-seg*), los que utilizan *modulación coherente*, y por otro lado los que utilizan *modulación diferencial*. Como en el caso de *gr-isdbt-tx* sólo se utiliza modulación coherente, los caminos que pueden tomar los segmentos son únicamente dos.

Posteriormente viene la etapa de entrelazamiento inter-segmentos. Precisamente consiste en intercalar las portadoras entre los segmentos que comparten la modulación. Para el segmento destinado a la recepción parcial este proceso no tiene sentido, por lo cual pasa directamente a la etapa de entrelazamiento intra-segmento.

El entrelazamiento intra-segmento se realiza permutando las portadoras de cada segmento entre sí. Se comienza por rotar las portadoras de acuerdo a la siguiente expresión:

Luego se realiza una aleatorización que está determinada por *Look Up Tables* que dependen del modo de transmisión y pueden consultarse en la referencia [?].

5.5.2. Entrelazamiento temporal

El entrelazamiento temporal consiste en distribuir los símbolos complejos entre distintos símbolos OFDM. Esta distribución o entrelazamiento se realiza para cada portadora, es decir que se realiza en el dominio del tiempo.

Esta técnica actúa como mecanismo de protección frente a ruidos impulsivos que típicamente se caracterizan por una corta duración en el tiempo. La profundidad o largo del entrelazamiento temporal es el parámetro que rige este proceso de entrelazado y puede ser seteado de manera independiente para cada capa. Está íntimamente ligado a la dispersión temporal que se realiza en los símbolos; a ma-

5.5. El uso de los entrelazamientos

por profundidad, los símbolos de una misma portadora son retardados un tiempo mayor.

La Figura 5.5 describe esquemáticamente el mecanismo de entrelazamiento. Consiste en aplicar un retardo distinto para cada símbolo dentro de un mismo segmento; los segmentos de una misma capa son retardados de igual manera según la profundidad de entrelazamiento elegida.

La implementación de estos retardos en *gr-isdbt-tx* es llevada a cabo por medio de colas en las que, de manera secuencial, se empujan los símbolos que van arribando correspondientes a las distintas capas jerárquicas. Desde el otro extremo se extrae un símbolo de cada cola también de manera secuencial.

Para una capa jerárquica con profundidad de entrelazamiento I_L , que puede ser 0, 1, 2, 4, 8, 16 según el modo, los retardos estipulados por ISDB-T para cada uno de sus segmentos se define de la siguiente manera:

$$q_L(i) = I_L \times m_i = I_L \times ((i \times 5) \bmod 96) \quad (5.1)$$

Donde $q_L(i)$ representa el tamaño de las colas o búfers; m_i es definido por el estándar como $m_i = (i \times 5) \bmod 96$, para $i = 0, 1, \dots, n_c$ con $n_c = 1, 2$, o 3 según el modo de transmisión.

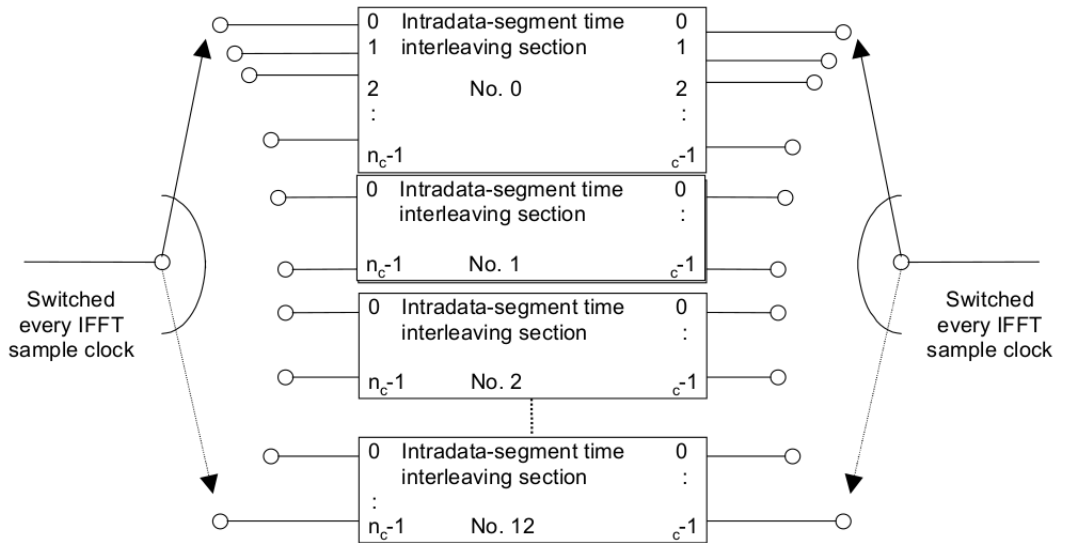


Figura 5.5: Mecanismo de entrelazamiento temporal.

El proceso de entrelazamiento introduce un retardo que debe ser ajustado de manera que el retardo total introducido por el proceso sea un múltiplo de cuadro OFDM. Para ello *gr-isdbt-tx* conforma colas más largas de las especificadas en el esquema de la Figura 5.5 logrando así un retardo total de un cuadro OFDM.

La cantidad de símbolos OFDM que genera de atraso el proceso está dada por:

$$95 \times I_L \bmod 204 \quad (5.2)$$

Capítulo 5. Un transmisor ISDBT implementado en GNU Radio

Entonces para completar un cuadro OFDM hacen falta $d_I = 204 - (95 \times I_L \bmod 204)$ símbolos OFDM. Por lo tanto la cantidad de cuadros OFDM de retardo será $N_L = (95 \times I_L + d_I) \bmod 204$.

La implementación en el transmisor de este ajuste de atraso consiste en empujar los símbolos que arriban al bloque a cada una de las colas de tamaño $I_L \times m_i + d_I$. Del otro extremo de las colas se toman los símbolos secuencialmente y se tiene la secuencia entrelazada. Este mecanismo, al igual que otros entrelazamientos, hace que en el arranque del sistema se tengan datos espurios en los registros.

5.6. Entrelazamiento de bits

Este bloque tiene a cargo esencialmente dos grandes tareas, primero entrelazar los bits que le ingresan provenientes de una determinada capa jerárquica dando lugar a un nuevo flujo de bits entrelazados. Luego estos bits son agrupados para formar símbolos complejos según la modulación que tenga la capa.

El proceso de entrelazamiento se lleva a cabo aplicando retardos definidos según la modulación. Esto supone utilizar una estructura de colas paralelas FIFO de distinto tamaño en las cuales se va colocando de a un bit a la vez. Una vez que se coloca un bit en cada cola, en el otro extremo se forma un complejo extrayendo el último bit de cada cola.

Inicialmente cuando se enciende el sistema se tendrá un transitorio con datos inválidos como producto del contenido de las colas previo al momento de que se inicializa el sistema. Esto no es inconveniente puesto que el receptor descarta esos datos y se sincroniza en un comienzo de frame válido.

Entrelazar utilizando esta mecanismo de colas de retardos variables implica agregar un retardo total correspondiente al camino más largo que deben atravesar los bits, es decir un retardo de 120 símbolos complejos. Según la modulación que utilice la capa estos 120 símbolos se traducen en un mayor o menor retardo en el tiempo; para QPSK corresponde a esperar que ingresen 120×2 bits en la entrada, mientras que para 64QAM deben pasar 120×6 bits.

Con el objetivo de tener un mismo retardo para lograr sincronismo en todas las capas se realiza un *ajuste de atraso*. En ISDB-T los retardos deben ser múltiplos de un símbolo OFDM; para el entrelazamiento de bits el retardo propio del proceso de entrelazamiento sumado al ajuste de atraso debe totalizar dos símbolos OFDM.

El tamaño de un símbolo OFDM corresponde a:

$$N_{OFDM} = m_L \times N_L \times 96 \times 2^{modo-1} \quad \text{bits} \quad (5.3)$$

N_{OFDM} depende de la modulación de la capa m_L , que puede ser 2, 4 y 6 para QPSK, 16QAM y 64QAM respectivamente; la cantidad de segmentos utilizados N_L y por supuesto del modo de transmisión. La solución implementada en *gr-isdbt-tx* consiste en incrementar los tamaños de las colas de manera que el retardo total sea de dos símbolos OFDM para cada capa. Por lo tanto para tener el retardo deseado los tamaños de las colas deben ser:

5.6. Entrelazamiento de bits

$$Q_i = q_i + \frac{2 \times N_{OFDM} - 120 \times m_L}{m_L} = q_i + N_L \times 96 \times 2^{modo-1} - 120 \quad (5.4)$$

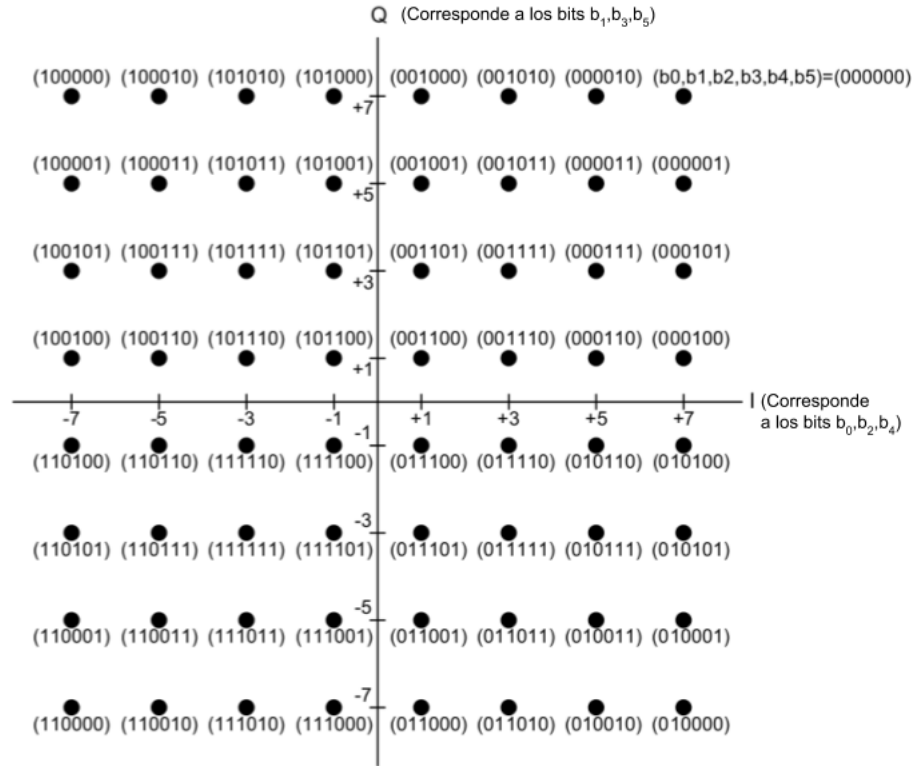


Figura 5.6: Mapeo de la constelación 64QAM.

Una vez que son empujados los bits en cada cola, desde el otro extremo se extrae un bit por cola y se mapean en un símbolo complejo de acuerdo a como lo establece la norma. Por ejemplo en la Figura 5.6 se muestra cómo se realiza este mapeo para la modulación 64QAM. Por último los complejos deben ser normalizados según un factor de normalización que se presenta en la Tabla 5.3.

Modulación	Factor de normalización
QPSK	$1/\sqrt{2}$
16QAM	$1/\sqrt{10}$
64QAM	$1/\sqrt{42}$

Tabla 5.3: Factores de normalización para los distintos esquemas de modulación.

Se trata de un conjunto de 204 símbolos OFDM, cada uno de ellos conformado por $13 \times 108 \times 2^{modo-1}$ símbolos complejos. Una de las particularidades del cuadro, es que su estructura tiene posiciones fijas, donde siempre viaja información con las características de la transmisión, y posiciones móviles, en las que viajan portadoras que estiman el efecto que tiene el canal sobre los datos, para poder, con esa información, robustecer al sistema.

Para cada segmento, la estructura del cuadro tiene su forma particular. En la siguiente figura presentamos un ejemplo para una modulación coherente, basada en un modo de transmisión 1.

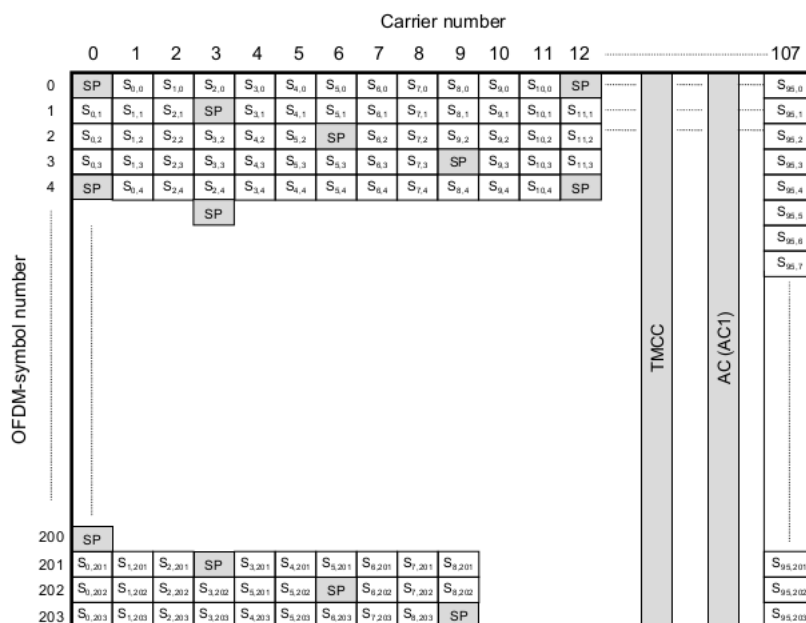


Figura 5.7: Estructura de cuadro OFDM para un segmento, con modo de transmisión 1.

A nivel de código, la solución que se implementó en este proyecto para el bloque, es bastante larga. Separamos el bloque en dos momentos, el transitorio y el régimen. Durante el transitorio, esto es, cuando se llama al bloque por primera vez y se crea en la memoria una instancia del mismo, se inicializan todas las variables. Entre ellas están, la palabra TMCC para todo el cuadro, los valores iniciales de la secuencia PRBS que rige el movimiento de las portadoras SP y algunas variables generales que mantienen conteos de frames para seguimiento.

En el caso general de un llamado a operar del bloque, la escritura se diseña organizada por simbolos OFDM. En cada llamado al bloque, se completara un simbolo generico. Algunas estructuras de control mantendran el control de los

5.8. La transformada de Fourier

cambios de palabras de control y otras variables. Los parametros genericos de transmision que dependen de parametros como el modo y las constelaciones de cada capa, se resuelven en el transitorio.

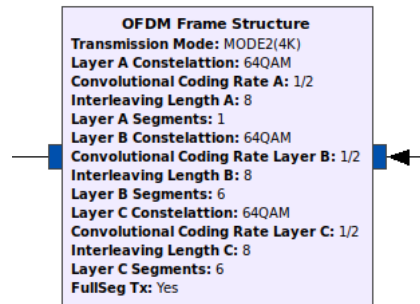


Figura 5.8: Bloque de conformacion de cuadro OFDM.

La cadena de sucesos de un llamado genérico es la siguiente. Primero se inicializa un vector de complejos en salida con el tamaño suficiente para todas las portadoras y los intervalos de guarda. Luego, se entra en una estructura de control que se encargara de repetir el proceso de escritura de símbolo OFDM cuantas veces el core de GNU Radio considere necesarios, a través del valor de *noutput-items*. Dependiendo del modo en el que se trabaje, se llama para cada segmento una función que se encarga de rellenar el segmento, según sea el modo de trabajo.

En caso de que se este trabajando en modo One-Seg, agregamos en este bloque un booleano que, en caso de estar en true, limita el rellenado de segmentos al segmento 0.

En el caso general, las funciones de llenado de segmento se llaman para todos los segmentos, en el orden lineal que tendran a la salida, el mismo que ya vimos en el capitulo 3.

Las funciones de llenado de segmento, toman como parámetro de entrada el numero de segmento que se va a escribir. Con este numero, encuentran en función de su posición en el cuadro, los indices en donde van las portadoras auxiliares y los datos. Luego iteran punto a punto por todas las portadoras del segmento, asignando según corresponda. En el caso de los datos, se copian simplemente de la entrada. Para el caso de las portadoras auxiliares, cada una tiene una funcion que almacena el estado de las mismas, y con ayuda de los indices de posicion y de simbolo dentro del cuadro OFDM, evolucionan los datos dinámicos de las mismas y escriben en el segmento el dato correspondiente.

5.8. La transformada de Fourier

Una vez formado el cuadro OFDM, de forma correcta con todas las portadoras de datos y auxiliares completas, es el momento de pasar la señal al dominio del tiempo, para agregarle el prefijo cíclico y ponerla en el aire.

Capítulo 5. Un transmisor ISDBT implementado en GNU Radio

Utilizamos para esto uno de los bloques de GNU Radio que implementan el algoritmo *Fast Fourier Transform*. Este algoritmo, que ya fue mencionado en el documento, permite realizar de forma rápida y suficientemente aproximada de la transformada de Fourier de una señal digital, siempre y cuando se maneje un tamaño de muestras que sea múltiplo de una potencia de 2.



Figura 5.9: Bloque que implementa el algoritmo FFT.

El parámetro *Reverse* del bloque, indica la dirección en la que se realiza la transformación. Dado que venimos trabajando en el dominio de la frecuencia y vamos a pasar al dominio temporal, debemos ejecutar el algoritmo en el sentido inverso.

5.9. El prefijo cíclico

Para sincronizar los relojes de transmisión y recepción, y eliminar de forma suficiente la interferencia intersimbólica, es que en ISDBT se utiliza un prefijo cíclico. Esto es, las primeras N muestras de la señal, se copian tal cual están hacia el final de la señal. Esto hace que, al calcular la autocorrelación de la señal en recepción, sea posible detectar de forma perfecta el comienzo de la señal, así como los parámetros de transmisión en caso de que sean desconocidos.

Se repasa el fundamento teórico detrás de esta práctica en el capítulo 3, en caso de buscar profundizar en la misma, una clara explicación de estos conceptos está en [6]. No entraremos en más detalles de este proceso para no ser repetitivos.

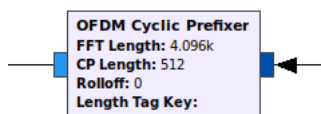


Figura 5.10: Bloque que implementa prefijo cíclico.

Este bloque tampoco es desarrollo propio del proyecto gr-isdbt-tx. Si bien escribimos un bloque que realiza esta funcionalidad (a nivel de programación se resuelve con un memcpy) preferimos dejar la implementación existente, que funciona muy bien y está validada por la comunidad de usuarios de GNU Radio.

De todos modos, la implementación de este bloque es un problema que los tutores nos plantearon al inicio del proyecto, y es interesante como primer acercamiento al entorno de desarrollo de GNU Radio. La lógica detrás del funcionamiento

del bloque es simple, por lo que el kid de la cuestión esta en manejar debidamente el movimiento de muestras entre el bloque y el core del software.

5.10. La transmision desde USRP

Como ya se discutió en secciones anteriores, para realizar la transmisión de la señal a través del equipo USRP, es necesario utilizar el bloque sink del complemento *gr-uhd*. Siempre que se mantengan las condiciones para la buena transmision que se enumeraron en el capitulo 4, es esperable tener a la señal en el aire sin mayores problemas.

Como consideraciones generales, recomendamos verificar que el espectro a utilizar este disponible, de forma de no solaparse con la señal de alguno de los canales comerciales, que transmiten a alta potencia y podrían causar problemas para decodificar nuestra señal.

Durante el desarrollo del transmisor, antes de probar la transmision contra un televisor comercial, una prueba bastante sencilla es la de transmitir hacia un archivo. Luego, podemos usar ese archivo como entrada de datos al receptor *gr-isdbt* para corroborar que la transmision tenga todos los parámetros correctos. Disponer de todos los bloques del receptor para tomar medidas, es un excelente método de debugging.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 6

Evaluación del sistema

Para la evaluación del transmisor, nos pusimos como objetivo final lograr la transmisión exitosa hacia un televisor comercial homologado por el LATU. En ese camino, nos propusimos una serie de objetivos de mas corto alcance. Desarrollamos en este capítulo los resultados de las pruebas realizadas con gr-isdbt-tx.

Primero, nos habíamos propuesto lograr la decodificación con gr-isdbt dentro del mismo flowgraph (Eso es, una prueba del sistema punta a punta trasmitiendo en condiciones ideales). Luego, nos propusimos una situación en la que una PC ejecuta el transmisor, enviando la señal a través del USRP, y otra computadora hace lo propio con el receptor gr-isdbt. Esto le agrega a la prueba anterior, el desafío de la puesta en el aire y decodificación de la señal ruidosa.

Finalmente, nos planteamos el objetivo final, la decodificación contra un televisor comercial. En este capítulo desarrollamos estas pruebas, y el nivel de éxito que alcanzo cada una.

6.1. Pruebas sobre gr-isdbt

Considerando que realizamos la mayoría del desarrollo contrastando los bloques del transmisor contra los de gr-isdbt, no parecería en principio un desafío mayor lograr la decodificación punta a punta. Pero en si, lo es, y eso es parte de la potencia del desarrollo con código abierto desde el otro lado.

Como receptor, gr-isdbt esta homologado por LATU, por lo tanto, podemos considerarlo como una simulación de un televisor comercial valida. Es en ese punto en el que comenzamos las pruebas. Dentro de un entorno controlado, el transmisor gr-isdbt-tx conectado a la entrada del receptor gr-isdbt.

Pruebas flowgraph

En este caso ideal, contenido en un mismo flowgraph y con nula variabilidad de las muestras, el transmisor se comporta de forma perfecta. Si encontramos, algunos problemas de consumo de CPU, seguramente debidos a problemas de optimización del código. Pero yendo estrictamente a las señales y a su procesamiento, entende-

Capítulo 6. Evaluación del sistema

mos que la decodificación correcta por parte de gr-isdbt, constituye un éxito para el transmisor.

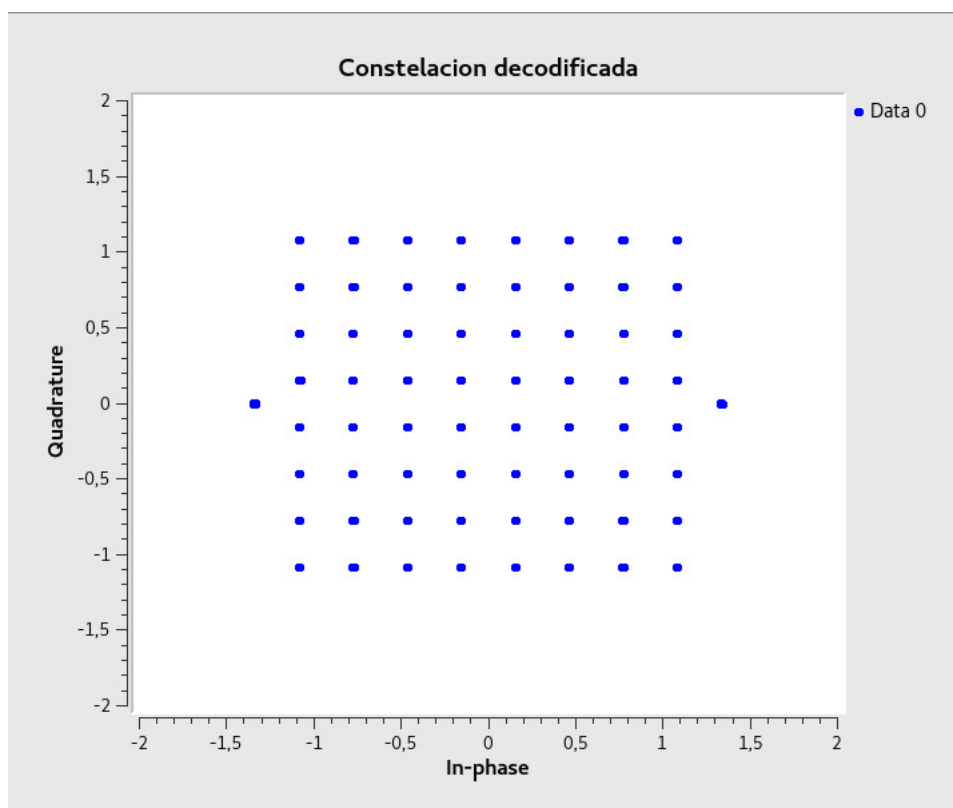


Figura 6.1: Constelación en recepción. Caso ideal.

Como se puede ver en la figura 6.1, el receptor gr-isdbt se sincroniza perfectamente con la señal transmitida, y en la constelación de recepción, vemos la misma constelación que en transmisión. Esto no debe sorprender, ya que en esta prueba, el canal está representado por un cable ideal.

Pruebas en el aire

Realizamos la misma prueba de antes, pero en lugar de simular un canal por un conector ideal del flowgraph de GNU Radio, utilizamos el aire y transmitimos en una línea vista, a través de 10 cm.

El resultado fue interesante. Si bien no es posible decodificar las señales todavía, intermitentemente es posible observar la constelación en recepción, como se ve en la figura 6.2.

Entendemos que tenemos algún problema de sincronismo, surgido seguramente por el uso del USRP. Dado que es el único cambio entre la prueba anterior y esta.

6.2. Pruebas sobre televisores comerciales

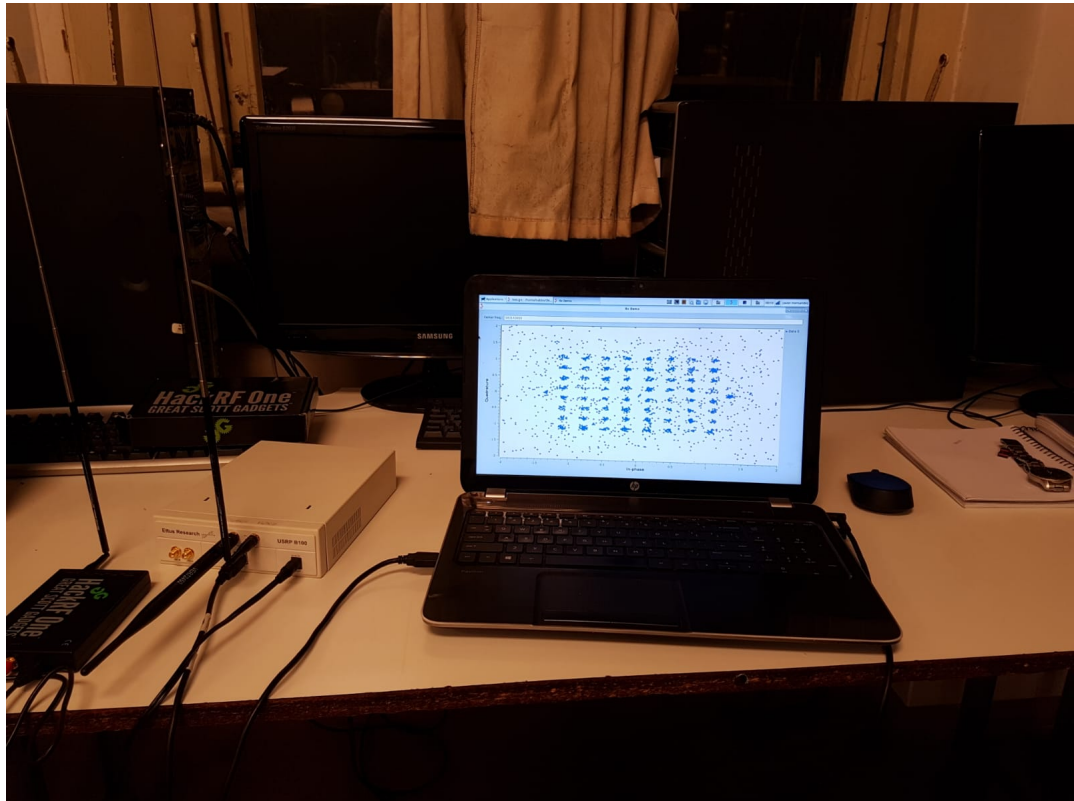


Figura 6.2: Constelación en recepción. Por línea vista.

6.2. Pruebas sobre televisores comerciales

Dado que las pruebas contra gr-isdbt con canal no ideal no fueron exitosas, no pasamos a las pruebas contra un televisor comercial. Hasta no resolver el problema de sincronismo encontrado en la prueba contra gr-isdbt usando el USRP como transmisor, no sera posible lograr el objetivo planteado al comenzar este proyecto.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 7

Conclusiones y trabajo a futuro

Este proyecto tenía como objetivos, dos grandes items. Por un lado buscábamos implementar un transmisor de televisión digital basado en SDR, cuya validez estaría determinada por la capacidad de transmitir hacia un televisor comercial homologado por LATU. Por otro lado, también buscábamos echar luz sobre el conocimiento disponible en el país sobre la norma ISDB-T.

Si bien entendemos que no fuimos capaces de cumplir con el primer objetivo, ese incumplimiento llega mas por un problema de plazos que por un problema de conocimientos. Logramos transmitir televisión digital en una situación ideal, lo cual no es menor. Los problemas existentes en gr-isdbt-tx para la transmisión por aire, entendemos que responden a errores en el manejo de los dispositivos USRP, y por ende, son solucionables.

En cuanto a este objetivo, entendemos que definitivamente no ha sido cumplido. Pero tampoco es justo decir que el grupo de trabajo fracaso en el objetivo. Puesto que dadas algunas condiciones de entorno, gr-isdbt-tx es capaz de transmitir televisión.

El funcionamiento del transmisor contra un televisor comercial, es cuestión de tiempo y de debugging. Por ende, consideramos que el objetivo principal del proyecto, se mantiene pendiente, a la espera de la resolución de los mencionados problemas.

Por otro lado, sobre el objetivo de aumentar la cantidad y la calidad de la información sobre la norma disponible para los técnicos nacionales, consideramos que este objetivo ha sido cumplido. El repositorio con el proyecto esta disponible de forma gratuita para cualquier interesado en aprender mas de la norma ISDB-T, y en conjunto con gr-isdbt, se tiene una herramienta de análisis del sistema, que cuenta con control absoluto de todas las variables.

Esperamos que este proyecto pueda colaborar con el mejor desempeño de la red nacional de transmisoras de televisión abierta, y que se hagan uso de las herramientas que la norma incluye.

Queda pendiente, para este grupo de trabajo, el mejor desarrollo de algunos de los bloques del sistema. Algunos items, como la portabilidad del divisor jerárquico, sera necesario resolver cuanto antes, para robustecer el funcionamiento del transmisor. Otros, como la ya mencionada resolución de los problemas de implementación

Capítulo 7. Conclusiones y trabajo a futuro

con el USRP, serán resueltos a la brevedad, no solo para dar cumplimiento al objetivo principal de este proyecto, sino también, para vencer la constante curiosidad que caracteriza a los ingenieros.

Finalmente, nos gustaría plantear un objetivo a mediano o largo plazo, que es un pequeño proyecto que hubiese sido implementado de disponer con el tiempo suficiente. Este transmisor, necesita como fuente de datos un archivo en formato MPEG TS. Sería muy interesante crear un conversor de MPEG 4 a transport streams, de modo que el transmisor pueda enviar al televisor cualquier video proveniente de la memoria del PC. Esta funcionalidad sería muy interesante y extendería el potencial del transmisor, ya que para este proyecto, obtener archivos en formato TS nos fue bastante complicado.

Referencias

- [1] ARIB. Association of radio industries and businesses, 1995. [Web; accedido el 06-11-2018].
- [2] Git User BogdanDIA. Dvb-t implementation in gnuradio, 2015. [Web; accedido el 06-11-2018].
- [3] Robert W Chang. Synthesis of band-limited orthogonal signals for multichannel data transmission. *Agregar Luego*, 1966.
- [4] Instituto Nacional de Estadística. *Encuesta Continua de Hogares*, 2017.
- [5] Gabriel Gómez Víctor González Barbone Pablo Belzarena Federico Larroca, Pablo Flores-Guridi. An open and free isdb-t full seg receiver implemented in gnu radio. *Mathematical Methods and Algorithms. Jhon Wiley and Son*, 2015.
- [6] Pablo Flores. La norma isdbt y un receptor implementado en sdr. 2016.
- [7] P. Flores-Guridi and F. Larroca. The isdb-t multiplex frame pattern explained. In *2017 IEEE URUCON*, pages 1–4, Oct 2017.
- [8] G David Forney. Concatenated codes. 1965.
- [9] Todd K Moon. Error correction coding. *Mathematical Methods and Algorithms. Jhon Wiley and Son*, 2005.
- [10] MPEG. Moving picture experts group, 1995. [Web; accedido el 06-11-2018].
- [11] Ettus Research. The usrpTM hardware driver repository, 2018. [Web; accedido el 24-11-2018].
- [12] Jan-Jaap Van de Beek, Magnus Sandell, and Per Ola Borjesson. Ml estimation of time and frequency offset in ofdm systems. *IEEE transactions on signal processing*, 45(7):1800–1805, 1997.
- [13] Weinstein and Edbert. Data transmission by frequency-division multiplexing using the discrete fourier transform. *Agregar Luego*, 1971.

Esta página ha sido intencionalmente dejada en blanco.

Índice de tablas

3.1. Parámetros relevantes en el estándar ISDB-T.	15
5.1. Parámetro indicador de capa en el IIP.	28
5.2. Bits por símbolo según modulación.	31
5.3. Factores de normalización para los distintos esquemas de modulación.	35

Esta página ha sido intencionalmente dejada en blanco.

Índice de figuras

1.1. Distribución de las estaciones transmisoras de TV digital en el Uruguay.	2
2.1. Esquema básico de codificación de canal.	7
3.1. Distribución espectral de los segmentos OFDM. A la derecha, entre el segmento 12 y el intervalo de guarda, puede apreciarse la presencia del piloto continuo.	13
3.2. Diagrama de bloques del transmisor ISDB-T definido por la ARIB.	15
3.3. Inserción del prefijo cíclico.	20
4.1. Arquitectura general de un equipo USRP	23
4.2. Bloque sink al equipo USRP	25
5.1. Bloque Reed Solomon Encoder de gr-dvb.	29
5.2. Circuito del código madre del Viterbi que implementa ISDBT. . .	30
5.3. Bloque de gr-dvb que implementa el codificador Viterbi.	30
5.4. Implementación de la modulación e interleaving a nivel de bits en gr-isdbt-tx.	31
5.5. Mecanismo de entrelazamiento temporal.	33
5.6. Mapeo de la constelación 64QAM.	35
5.7. Estructura de cuadro OFDM para un segmento, con modo de transmisión 1.	36
5.8. Bloque de conformación de cuadro OFDM.	37
5.9. Bloque que implementa el algoritmo FFT.	38
5.10. Bloque que implementa prefijo cíclico.	38
6.1. Constelación en recepción. Caso ideal.	42
6.2. Constelación en recepción. Por línea vista.	43

Esta es la última página.
Compilado el lunes 1 octubre, 2018.
<http://iie.fing.edu.uy/>