

MÁSTER UNIVERSITARIO EN CIENCIA Y TECNOLOGÍA ESPACIAL

TRABAJO FIN DE MÁSTER

SPARSE VECTOR CODES PARA COMUNICACIONES ESPACIALES



Estudiante: Bellis García, Daniel Owen

Director: Angueira Buceta, Pablo

Curso: 2022-2023

Fecha: Bilbao, 19, Julio, 2023

Sparse Vector Codes para Comunicaciones Espaciales

Autor:

Daniel Owen Bellis García

Director:

Pablo Angueira Buceta

Grupo de Tratamiento de la Señal y Radiocomunicaciones

Máster en Ciencia y Tecnología Espacial

BILBAO, Julio 2023

*El cosmos es todo lo que es, todo lo que fue y todo lo que será.
Nuestras más ligeras contemplaciones del cosmos nos hacen estremecer:
sentimos como un cosquilleo nos llena los nervios, una voz muda, una ligera
sensación como de un recuerdo lejano o como si cayéramos desde gran altura.
Sabemos que nos aproximamos al más grande de los misterios.*

Carl Sagan.

Agradecimientos

Este Trabajo Fin de Máster es el resultado de varios meses de trabajo y aprendizaje. Quiero expresar mi agradecimiento, en primer lugar, a mi director, Pablo, por brindarme la oportunidad de desarrollar mi trabajo en TSR y por guiarme en su realización. Además, quiero agradecer a Iñigo y a Simon por su apoyo y sus consejos, que me han ayudado a superar los obstáculos que se me han ido presentando a lo largo del desarrollo del TFM. Quiero hacer especial mención a todos mis compañeros del grupo TSR, quienes me han acogido y compartido conmigo estos meses de trabajo, en especial a Filip, a Teresa y a Mash. Sin ellos esta experiencia no hubiera sido lo mismo.

Me gustaría también agradecer a mis compañeros del Máster en Ciencia y Tecnología Espacial, con quienes he compartido aula durante este curso, y quienes forman un grupo excepcional. Además, quiero agradecer a Germán, a Jose y a Pepe por todas las experiencias y momentos vividos juntos, y los que quedan...

Por último, quiero agradecer a mis padres y a mi hermano, quienes nunca han dejado de apostar por mi y han sido un apoyo incondicional para lograr todas mis metas y sueños.

Resumen

En el ámbito de la exploración espacial, las comunicaciones inalámbricas son de vital importancia, y de ellas depende en muchos casos, el éxito de una misión. Debido a las grandes limitaciones, en cuanto a recursos, que impone el entorno espacial, las técnicas de codificación contra errores suponen una gran ayuda para la optimización de las comunicaciones. Permiten la recepción de mensajes a pesar de la poca potencia disponible y el ruido introducido, lo cual sería imposible sin hacer uso de este tipo de técnicas. Muchas de estos códigos ya están siendo utilizados para misiones espaciales, pero a pesar de que son eficientes para mensajes largos, pierden eficacia a medida que la longitud de mensaje disminuye. Por ello, es importante introducir nuevos códigos que sean óptimos para el envío de mensajes cortos, como son los SVC (*Sparse Vector Coding*).

Los SVC son unos códigos de corrección contra errores optimizados para la transmisión de mensajes cortos. En 2018, Hyung-ju et al. presento a los SVC como una alternativa a las actuales propuestas de codificación. Este trabajo pretende partir de TFM presentado por Iñigo Bilbao, en el cuál realiza un estudio más concluyente, validando los resultados presentados por la bibliografía, aportando más conocimiento acerca de su comportamiento general y proponiendo una arquitectura mediante una plataforma de simulación. Este TFM concentra sus esfuerzos en la etapa de decodificación del sistema, ya que es uno de los puntos clave para optimizar su rendimiento, y busca plantear alternativas para conseguir este objetivo.

Laburpena

Espazio-esplorazioko esparruan haririk gabeko komunikazioak garrantzia handikoak dira eta misio bat arrakastatsua izan dadin nahitaezkoak dira. Espazioko baliabide faltak eragindako mugak direla eta, errorean aurkako kodifikazio-teknikek komunikazioen optimizazioa asko errazten dute. Potentzia gutxi eskuragarri izan arren eta zarata izan arren mezuen bidalketa eta jasotzea ahalbidetzen dute. Teknika hauek gabe aurrekoa ezinezkoa litzateke. Honelako kode asko espazio-misioetan erabiltzen dira, baina mezu luzeetarako egokiak diren arren, mezu laburrekin ez dira hain eraginkorrak. Hortaz, SVC (Sparse Vector Coding) bezalako mezu laburrekin kode optimoak garatu behar dira.

SVC kodeak mezu laburrak bidaltzeko eta jasotzeko optimizatutako kodeak dira. 2018an Hyung-ju eta abarrek gaur egun erabiltzen diren kodifikazio tekniken alternatibo gisa SVC aurkeztu zuten. Lan honek Iñigo Bilbaok aurkeztutako MALetik abiatzen da. Bertan Iñigok ikerketa sakon bat egiten du, bibliografian aurkezten diren emaitzak balidatu egiten ditu, eta horretaz gain portaera orokorrari buruzko jakintza gehitzen du eta simulazio-plataforma baten bitartez arkitektura bat proposatzen du. MAL honetako arreta handiena sistemaren deskodifikazio-etapan jarri da, errendimendua hobetzeko gakoetariko bat baita eta helburu hau lortzeko aukera desberdinak aurkezten ditu.

Summary

In the field of space exploration, wireless communications are of a vital importance, and the success of a mission depends on them in many cases. Due to the great limitations, in terms of resources, imposed by the space environment, error-correcting coding techniques are a great help for the optimization of communications. They allow the reception of messages despite the low power available and the noise, which would be impossible without using this type of techniques. Many of these codes are already being used for space missions, but although they are efficient for long messages, they lose effectiveness as the message length decreases. Therefore, it is important to introduce new codes that are optimal for sending short messages, such as SVC (Sparse Vector Coding).

SVCs are error-correcting codes optimized for the transmission of short messages. In 2018, Hyoung-ju et al. presented SVCs as an alternative to current coding proposals. This study takes the TFM by Iñigo Bilbao as a starting point. In his project, he carries out a more conclusive study, validating the results presented by the bibliography, providing more knowledge about its general behavior, and proposing an architecture through a simulation platform. This TFM focuses on the decoding stage of the system, since it is one of the key points to optimize its performance and, seeks to propose alternatives to achieve this goal.

Índice

1. Introducción	5
Antecedentes	5
Motivación	5
2. Objetivos	8
3. Metodología	9
Descripción de fases	9
Diagrama de Gantt	9
4. Beneficios	10
Beneficios técnicos del trabajo.....	10
Contribución a los ODS.....	10
5. Estado del arte.....	11
Estudio de temas básicos de codificación de canal.....	11
Sparse Vector Coding	13
6. SVC en las comunicaciones espaciales	18
Estándares de Envío de Datos	20
7. Algoritmo de decodificación	22
Rendimiento	25
Complejidad	28
Caso de uso	30
8. Conclusiones.....	32
9. Bibliografía.....	33

Índice de figuras

Figura 1. Diagrama de Gantt.	9
Figura 2. ODS 9.	10
Figura 3. Curvas BLER para codificación contra errores de mensajes cortos [12].	12
Figura 4. Esquema SVC [13].	13
Figura 5. Clasificación de los algoritmos CS [15].	15
Figura 6. Rendimiento y velocidad de los algoritmos Greedy [18].	16
Figura 7. El segmento espacial para una red de satélites de comunicaciones [14].	18
Figura 8. Deep Space Network (DSN), NASA.	19
Figura 9. Contexto operaciones de la DSN [16].	21
Figura 10. Algoritmo de búsqueda MMP.	22
Figura 11. Algoritmo de búsqueda MMP-DF.	23
Figura 12. Validación sistema bibliografía [5].	25
Figura 13. Curva BLER para $R=1/2$	27
Figura 14. Curva BLER para $R=1/4$	27
Figura 15. Curva BLER MMP-DF, para diferentes R.	28
Figura 16. Curva BLER para envío de comandos.	31

Índice de tablas

Tabla 1. Rendimiento y velocidad de los algoritmos CS [15].....	16
Tabla 2. Códigos de corrección contra errores usados en diferentes misiones [7].....	19
Tabla 3. Codificación y margen del enlace para diferentes tamaños de bloque [16].	20
Tabla 4. Parámetros sistema comunicaciones.....	25
Tabla 5. Medidas de complejidad del algoritmo Breadth-First para diferentes lenguajes de programación.	29
Tabla 6. Medidas de complejidad del algoritmo Depth-First para diferentes lenguajes de programación.	29

1. Introducción

Antecedentes

Desde los años 50, con el lanzamiento de *SPUTNIK I* por la antigua URSS, la humanidad se ha visto inmersa en un viaje hacia la exploración fuera de las fronteras de nuestro planeta. De forma paralela y gracias a la inversión y motivación por explorar nuestro Sistema Solar y más allá, se han desarrollado muchas tecnologías que se hubieran demorado en el tiempo. Uno de los campos clave para el éxito de las misiones espaciales es el campo de las comunicaciones inalámbricas o a distancia. Este tipo de comunicaciones están basados en el principio físico de la transmisión de ondas electromagnéticas que se propagan en el espacio libre. En el año 1895, fue *Guglielmo Marconi* quien abrió el camino para las comunicaciones inalámbricas modernas al transmitir el código Morse de tres puntos para la letra 'S' a una distancia de tres kilómetros mediante ondas electromagnéticas. A partir de ello, las comunicaciones inalámbricas se han convertido en un elemento que ha provocado un cambio profundo en la sociedad, y que además ha permitido poder lanzarse hacia la exploración fuera de nuestro planeta [1]. En el ámbito espacial, es de vital importancia mantener un contacto continuado con las naves o satélites, para poder monitorizar su estado funcional, poder mandar comandos para realizar correcciones de trayectoria o de actitud, recibir datos de los diferentes instrumentos dedicados a ciencia, etc... Estas operaciones requieren una gran robustez en las comunicaciones, la cual se puede obtener diseñando una antena de gran tamaño para la nave o dimensionando un sistema de transmisión que aporte una mayor potencia. Estas medidas afectan en gran medida al diseño mecánico de la nave, siendo necesaria una fuente de energía que sea capaz de alimentar la potencia al sistema de transmisión y una mayor masa en hardware del sistema (incluyendo antenas y el sistema de transmisión), por lo que conseguir una mayor robustez implica una gran complicación tecnológica además de un mayor coste. Según la revista *Business Insider* el coste por masa para lanzar objetos al espacio ronda los 60.000 dólares por kg [2]. En este contexto, los sistemas de codificación contra errores surgen como una alternativa para mejorar la robustez de las comunicaciones.

Motivación

A partir del contexto planteado en el apartado anterior, se proponen los códigos de codificación contra errores como una alternativa que puede ayudar a la hora de ahorrar en cuanto a recursos y costes. En el sector espacial esto puede ser un factor clave para el éxito de una misión, ya que se necesita de robustez a la hora de enviar y recibir mensajes en entornos en la que la relación entre la señal y el ruido recibidos (SNR) es muy baja. En muchas de estas misiones, como la misión de la *NASA OSIRIS-Rex* [3], se emplea un transpondedor para las comunicaciones desde la Tierra con el espacio profundo y viceversa denominado *Small Deep Space Transponder* (SDST). Este dispositivo incorpora códigos convolucionales para la codificación contra errores de los mensajes [4]. A la hora de emplear este tipo de técnicas de codificación surge una problemática asociada a la longitud de los mensajes, ya que son unos códigos que no están

optimizados para el envío de mensajes cortos. Debido a la limitación en potencia que existe desde el espacio para generar las ondas de radiofrecuencia, se opta por el envío de mensajes cortos en aplicaciones esenciales como la monitorización de la nave mediante telemetría y telecontrol para que, de esta forma, se envíen con la máxima potencia. La codificación contra errores se encuentra ampliamente desarrollada para el envío de mensajes largos, pero pierde capacidad de corrección a medida que las longitudes de mensaje disminuyen. Por ello y por los muchos sectores en los que existe una gran aplicabilidad para la codificación contra errores para mensajes cortos, como en entornos industriales, para el 5G o el Internet de las Cosas, se ha realizado un esfuerzo en los últimos años para poder optimizar los códigos de corrección para mensajes cortos.

En este sentido, se presentaron unos nuevos códigos en 2018 que pretendían competir junto con otros códigos como los polares o LDPC, para optimizar la codificación contra errores para el envío de mensajes cortos. Los denominados códigos SVC (*Sparse Vector Coding*), optimizados para la codificación y decodificación de mensajes cortos, surgen como una alternativa que además de ofrecer una buena capacidad de corrección de errores, trabajan con baja latencia, es decir, ofrecen una codificación y decodificación rápida [5]. Como todas las técnicas de codificación, esta se basa en añadir bits de redundancia al mensaje que se pretende mandar a través de un canal ruidoso, que será dado y perturbado por el ruido y deberá ser reconstruido en recepción. El proceso de codificación se realizará con palabras de la misma longitud a en el caso de SVC, por lo que habrá una limitación de 2^a palabras de mensaje posibles. La codificación de estas palabras de mensaje se puede dividir en dos etapas: el “mapeo a un vector k -disperso” y la “conversión al mensaje codificado”. La primera consiste en asignar a cada palabra de mensaje un vector formado por k valores no nulos (*sparse vector*), siendo k mucho menor que la longitud total del vector. Por lo tanto, cada palabra de mensaje se traduce a un vector que posee pocos valores no nulos en comparación con su longitud total. En la segunda etapa, el vector disperso asignado a la palabra de mensaje se utilizará para obtener una combinación lineal de k palabras de código preestablecidas, donde k es el número de valores no nulos del vector disperso [5]. Estas palabras de código tendrán una longitud mucho mayor a las palabras de mensaje, por lo que en este proceso se habrán añadido bits de redundancia. Los mensajes codificados serán enviados a través del canal, que puede incorporar ruido y corromper el mensaje. Para la decodificación, se hará uso del *Compressive Sensing* (CS) [6] una técnica con base matemática que permite solucionar sistemas de ecuaciones indeterminados cuando uno de los vectores tiene pocos valores no nulos. A partir del mensaje corrupto se puede recuperar el mensaje original utilizando algoritmos desarrollados para CS. Estos algoritmos permiten recuperar el vector k -disperso, a partir del cual es inmediato recuperar la palabra de mensaje.

El principal problema de este código es que no existe un estudio suficientemente profundo (además de que la mayor parte de la bibliografía está escrita por los mismos autores) que muestre de una manera clara su capacidad potencial para la corrección de errores. En este sentido hay dos principales parámetros que se podrían estudiar para poder optimizar este sistema de codificación y alcanzar su potencial: la matriz de codificación y el algoritmo de decodificación. En anteriores estudios se planteó un caso “estándar” [5] [7] en el que se empleaba una matriz de codificación con entradas aleatorias de Bernoulli (1 o -1 con la misma probabilidad) y un algoritmo de decodificación basado en *Multipath Matching Pursuit* (MMP) (ambos se explicarán en los apartados posteriores).

Con el presente trabajo se pretende realizar un estudio del estado del arte existente sobre los SVC hasta el momento, contribuyendo en aportar una alternativa más eficiente para el algoritmo de decodificación, evaluándola mediante una plataforma de software en la que se simule un sistema de comunicaciones empleando esta codificación. Si pretende realizar un estudio comparativo del rendimiento respecto a anteriores implementaciones [5] [7], y un análisis comparativo de la complejidad de los algoritmos. Además, se estudiará su posible uso en el ámbito de las comunicaciones espaciales.

2. Objetivos

El objetivo del presente Trabajo Fin de Máster (TFM) es contribuir al diseño de una arquitectura de sistema de comunicaciones inalámbricas para el envío de mensajes cortos utilizando los códigos SVC como sistema de protección contra errores. Se busca de forma específica un algoritmo de decodificación en el receptor que mejore o amplie el rendimiento de los utilizados anteriormente [5] [7]. Se comenzará por un estudio del estado del arte de los códigos de corrección de errores para mensajes cortos SVC [5]. Se realizará un análisis desde el ámbito de las comunicaciones espaciales, ya que dentro de este marco es característico el envío de mensajes cortos (telemetría, telecomando, etc...), y se podría optimizar de forma sustancial el envío de información ya que los códigos empleados actualmente están optimizados para mensajes largos. Es por ello que los SVC constituyen una alternativa a tener en cuenta para el futuro de las comunicaciones espaciales.

Además, el objetivo principal del TFM es presentar un **algoritmo de decodificación** alternativo y más eficiente a los planteados hasta ahora. Para ello se realizará un análisis del rendimiento y la complejidad del nuevo algoritmo, comparándolo con los propuestos por la bibliografía [5] [7].

Para conseguir este objetivo, el trabajo se ha dividido en varias metas intermedias:

- Análisis del estado del arte de la codificación Sparse Vector Coding.
- Análisis del estado del arte de los algoritmos de decodificación.
- Implementación y validación de un sistema de comunicaciones que incluya un algoritmo de decodificación alternativo y realizar un análisis comparativo con el sistema implementado en la bibliografía.
- Estudio del estado del arte de estándares/sistemas de telemetría y comandos en aplicaciones de espacio.
- Diseño de un formato de telemetría que incluya SVCs como sistema de codificación.

3. Metodología

Descripción de fases

A continuación, se muestra la metodología (fases) que se ha seguido para poder cumplir con los objetivos propuestos en el apartado anterior:

- Fase 1. Estudio del estado de arte: estudio del estado del arte de la codificación SVC hasta la fecha.
- Fase 2. Estudio del estado del arte de los algoritmos de decodificación: se han estudiado las posibles familias de algoritmos planteadas por la bibliografía, para poder analizar que algoritmos podrían aplicarse al problema del CS.
- Fase 3. Estudio de los SVC en las comunicaciones espaciales: estudio de las comunicaciones espaciales orientado a la corrección de errores y el interés de los SVC en este ámbito. Estudio del estado del arte de estándares/sistemas de telemetría y comandos en aplicaciones de espacio.
- Fase 4. Análisis del rendimiento: una vez encontrado un posible candidato, se ha realizado un estudio implementando una plataforma de simulación de un sistema de comunicaciones que incorpora el algoritmo propuesto, y realizando un estudio comparativo con el algoritmo planteado por la bibliografía. De esta forma se puede verificar el rendimiento del algoritmo.
- Fase 5. Análisis de complejidad: se ha procedido a buscar posibles métodos para hacer un análisis de su complejidad. Una vez encontrado el método adecuado, se puede realizar un análisis comparativo con el algoritmo planteado por la bibliografía.
- Fase 6. Caso de uso espacial: se plantea un caso de uso de la plataforma de simulación empleada para el caso anterior aplicado al formato de telemetría.
- Fase 7. Redacción de la memoria: redacción del documento final que constituye el Trabajo Fin de Máster.

Diagrama de Gantt

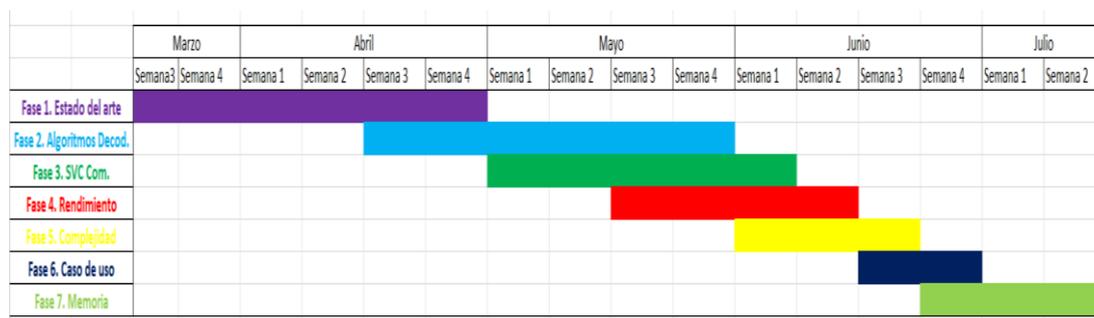


Figura 1. Diagrama de Gantt.

4. Beneficios

Beneficios técnicos del trabajo

Este trabajo busca contribuir de forma amplia en el apartado de decodificación al diseño de una arquitectura de sistema de comunicaciones inalámbricas para el envío de mensajes cortos utilizando los códigos SVC como sistema de protección contra errores. Esto supondría un gran avance para la optimización de este tipo de códigos para longitudes de mensaje más cortas, ya que la eficacia de los códigos más empleados hasta la fecha se ve reducida a menor tamaño de mensaje. De esta forma se contribuiría a presentar un sistema fiable a la bibliografía de los SVC y de los métodos de codificación para mensajes cortos, el cual podría estudiarse para futuras implementaciones.

Contribución a los ODS

En 2015, la ONU aprobó la Agenda 2030 sobre el Desarrollo Sostenible, una oportunidad para que los países y sus sociedades emprendan un nuevo camino con el que mejorar la vida de todos, sin dejar a nadie atrás. La Agenda cuenta con 17 Objetivos de Desarrollo Sostenible, que incluyen desde la eliminación de la pobreza hasta el combate al cambio climático, la educación, la igualdad de la mujer, la defensa del medio ambiente o el diseño de nuestras ciudades [6]. Los objetivos del presente TFM se pueden enmarcar dentro del objetivo de desarrollo sostenible número 9: “INDUSTRIA, INNOVACIÓN E INFRAESTRUCTURA”.



Figura 2. ODS 9.

En este sentido, se puede englobar en el objetivo 9 ya que, con este TFM, sectores en los que son necesarios los mensajes cortos con baja latencia como es el industrial, para las comunicaciones inalámbricas industriales mediante mensajes cortos, podría verse muy beneficiados, junto con otras aplicaciones como el 5G. Además, supondría una nueva técnica de codificación contra errores que podría mejorar la eficacia de las actuales técnicas empleadas en el sector espacial para las comunicaciones.

5. Estado del arte

A la hora de plantear el contexto y antes de desarrollar la teoría que comprenden los SVC es útil dar un primer paso desde el planteamiento general de un sistema de comunicaciones. De forma general, los sistemas de comunicaciones están formados por un emisor que emite el mensaje, el canal por el cual se manda el mensaje y el receptor que recibe el mensaje enviado a través del canal. Debido a las perturbaciones que puede sufrir el mensaje debido al ruido que puede introducir el canal, hay que asegurarse de que el mensaje sea recibido correctamente por el receptor. En este sentido, se emplea una codificación contra errores al mensaje que es enviado para de esta manera minimizar la pérdida de información, y recibir tras decodificar el mensaje en recepción de forma correcta. Por lo tanto, usar este tipo de técnicas es muy útil para un uso eficiente de las comunicaciones inalámbricas.

Estudio de temas básicos de codificación de canal.

La codificación contra errores es un método efectivo para aumentar la robustez de la transmisión de información. Su objetivo es transmitir la máxima cantidad de información posible por un canal que introduce ruido y distorsión y, por tanto, puede degradar y modificar el mensaje, consumiendo los menores recursos posibles. La robustez de la señal se puede medir mediante la “Signal to Noise Ratio” (SNR), que mide la intensidad de señal frente a la del ruido. Se puede conseguir una mejor SNR si se envía el mismo bit de mensaje n veces, pero esto afectará a la velocidad de envío de la información, tardando n veces más en recibir el mensaje, pero con una mayor probabilidad de recibir el mensaje correcto. Por tanto, se busca el compromiso entre robustez, es decir capacidad de corrección contra errores y el consumo de recursos, o la velocidad en la transmisión de información (latencia) [7].

En 1948, *C.E. Shannon* a partir de los trabajos de *Nyquist* y *Hartley* publica el artículo de referencia en el campo de la codificación contra errores en el que define el concepto de “capacidad de canal” o “límite de Shannon” que relaciona la cantidad máxima de información que se puede transmitir por un canal para un número de bits de mensaje concreto. A partir de este artículo, se ha intentado aproximarse con diferentes técnicas de codificación a este límite teórico para poder optimizar las comunicaciones en robustez y recursos [9].

Se pueden diferenciar dos grupos de códigos en este contexto [10]:

- Códigos de bloque: el mensaje es dividido en palabras de mensaje de longitud b bits. Posteriormente, estas palabras de código se mapean a palabras de código de longitud m bits. Estas palabras de código serán las que se enviarán por el canal ruidoso por lo que se habrá añadido redundancia respecto a las palabras de mensaje ($m > b$). Para cuantificar esta redundancia podemos hacer uso del “Code Rate” (1), que es la relación entre las longitudes de palabras de mensaje y las de código:

$$R = \frac{b}{m} \tag{1}$$

- **Códigos convolucionales:** Los códigos convolucionales son códigos lineales, donde las palabras de mensaje también son mapeadas a palabras de código, al igual que en los códigos de bloque. Sin embargo, tienen memoria, por lo que la codificación actual depende de los datos que se envían en el presente y que se enviaron en el pasado. El reto de estos códigos es saber optimizar la memoria utilizada para conseguir la mayor corrección de errores posible.

Para poder medir la eficiencia de los códigos de corrección de errores, se realizan simulaciones en las que se envían mensajes aleatorios. Estos mensajes se codifican al ser enviados, atraviesan el canal con ruido y se decodifican en recepción. Posteriormente, se puede contar la cantidad de palabras que han sido recibidas correctamente y se puede obtener la tasa binaria de error, que se mide mediante el BLER (*Block Error Rate*). La BLER se medirá para diferentes niveles de relación señal a ruido obteniendo las “curvas BLER” [Figura 2]. Así se puede caracterizar la capacidad de corrección de diferentes tipos de código, y es el método que seguiremos en apartados posteriores para los SVC.

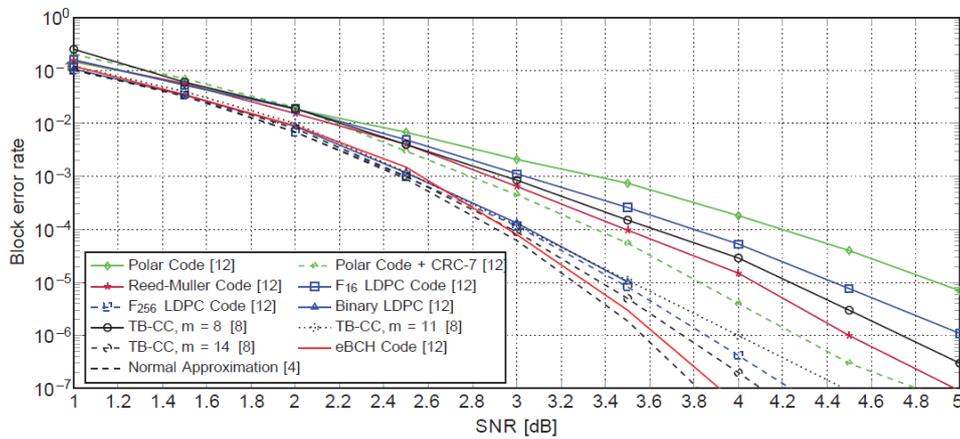


Figura 3. Curvas BLER para codificación contra errores de mensajes cortos [12].

En la actualidad, se han desarrollado códigos como los LDPC (*Low Density Parity Check*) que se acercan mucho al límite de Shannon [11]. Sin embargo, estos códigos están optimizados solo para cuando los mensajes enviados sean lo suficientemente largos y su eficiencia cae a medida que se disminuye su longitud [12]. Por ello y por los muchos sectores en los que existe una gran aplicabilidad para la codificación contra errores para mensajes cortos, como en entornos industriales, para el 5G o el Internet de las Cosas, se ha realizado un esfuerzo en los últimos años para poder optimizar los códigos de corrección para mensajes cortos. Es en este contexto en el que se proponen los “*Sparse Vector Codes*” (SVC) [5].

Sparse Vector Coding

La nueva técnica de codificación “Sparse Vector Coding” fue presentada en 2018 por Hyoungju et al. [5]. Es un código de corrección de errores diseñado y optimizado para el envío de mensajes cortos, y que se basa en el mapeo a un vector disperso y al uso del “Compressive Sensing” (CS) [6] en recepción. Gracias a los algoritmos de CS se puede realizar una decodificación optimizada en cuanto a recursos temporales. A continuación, se muestra una figura del esquema de los SVC:

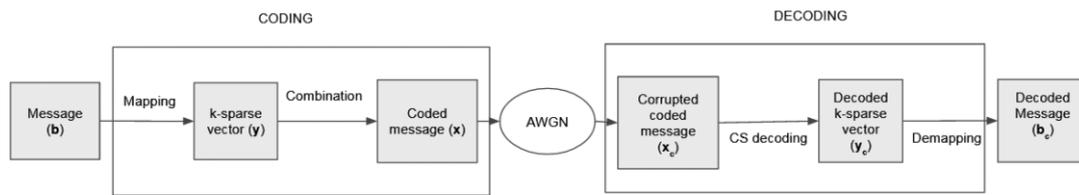


Figura 4. Esquema SVC [13].

La primera etapa que constituye la codificación, como se ve en la [Figura 3], consta de 2 partes: el “mapeo a un vector k-disperso” y la “conversión al mensaje codificado”. Antes de pasar por el codificador, el mensaje será dividido en palabras de mensaje de longitud b . Estas palabras de mensaje serán mapeadas a un vector k-disperso con muy pocos valores no nulos (k valores no nulos). De esta forma, a cada mensaje posible de longitud b le corresponderá un vector k-disperso único (total de 2^b mensajes diferentes y vectores k-dispersos asociados). Por lo tanto, se puede deducir que la longitud de los vectores k-dispersos será mucho mayor al de las palabras de mensaje ($n \gg b$). Estos vectores tendrán una longitud muy grande y consumirán muchos recursos en el caso de enviarlos, por lo que hay una segunda fase de la codificación, la “conversión al mensaje codificado”. En esta fase se emplea un diccionario de palabras de código (una matriz), cuyas palabras de código deben cumplir ciertas propiedades para que sean compatibles con CS. Con cada vector k-disperso se creará una combinación lineal de palabras de código, que constituirá el mensaje codificado. Este mensaje se enviará por el canal ruidoso (se realizará la simulación en apartados posteriores con ruido blanco, AWGN), y se iniciará el proceso de decodificación en recepción usando algoritmos desarrollados a partir de CS. Estos algoritmos deben reconocer el mensaje codificado deteriorado, que en este caso se podrá asociar a un vector k-disperso. Por último, se podrá recuperar cada palabra de mensaje original a partir de cada vector k-disperso asociado.

Es importante analizar las características y condiciones que impone el CS [6] [14]. El problema establece que siendo una señal k-dispersa y ($k \ll n$), una señal con pocos valores no nulos (k) en comparación con su longitud total (n), se puede recuperar la señal midiendo unos pocos ($m < n$) valores de la señal. Estas medidas se realizarán mediante la matriz C :

$$Cy = x \quad (2)$$

La matriz C tendrá unas medidas de $m \times n$, y será el vector k-disperso y x el vector de medidas reducidas de dimensión m . Por tanto, x será una combinación lineal de columnas de la matriz C . El sistema está indeterminado, con un mayor número de incógnitas n que ecuaciones m ($m <$

n). Sin embargo, al ser el vector y k -disperso, conocer las posiciones de los valores no nulos del vector disperso supondría colapsar el sistema indeterminado a uno sobredeterminado de k incógnitas y m ecuaciones ($k < m$). Este es el objetivo principal del CS.

El proceso sería sencillo si las columnas de la matriz C fueran ortonormales entre sí, ya que supondría que el vector x es una combinación lineal de columnas de C :

$$x = \alpha c_1 + \beta c_2 + \dots + \omega c_3 \quad (3)$$

donde los coeficientes α, β y ω son los valores del vector disperso. Para conocer su valor, simplemente hay que multiplicar una a una las columnas de la matriz con el vector. Al ser las columnas de la matriz ortonormales entre sí, la operación es una isometría y , por tanto, cumple el teorema de Pitágoras:

$$\|Cy\|_2^2 = \|y\|_2^2 \quad (4)$$

Este teorema se aplica a bases ortonormales. Hay que tener en cuenta que para un espacio de dimensión m , solo pueden conseguirse m vectores ortogonales, por lo que la matriz solo puede tener dimensiones $m \times m$ como máximo. Sin embargo, para el caso de CS, la matriz de medida C posee unas dimensiones $m \times n$ siendo $m < n$, por lo que no se puede imponer que las columnas sean completamente ortonormales.

Como solución, en [6] se propone una generalización del teorema de Pitágoras para matrices no cuadradas, el RIC (*Restricted Isometry Property*). Para que una matriz C cumpla la condición de RIC de orden δ_k , la matriz debe cumplir la ecuación (5):

$$(1 - \delta_k)\|y\|_2^2 \leq \|C_k y\|_2^2 \leq (1 + \delta_k)\|y\|_2^2 \quad (5)$$

siendo C_k la submatriz al escoger k columnas de la matriz C , y la señal que se quiere medir y δ_k la RIC, la más pequeña que cumpla esta condición. Además, $\|\cdot\|_2$ indica la norma euclidiana. En la actualidad, no existen métodos analíticos para conseguir matrices con RICs suficientemente bajas. Por lo tanto, el método más utilizado es el de generar estas matrices de manera aleatoria con entradas independientes de Bernoulli o Gaus para cada valor de la matriz. Este método, es el que ha sido empleado por Hyoungh-ju et al. [5], e Iñigo Bilbao [13] en sus respectivos estudios, y es el que se usará en el presente TFM.

Además de esto, cabe destacar que las medidas habrán sido corrompidas por el ruido v y, por tanto, se puede reescribir la ecuación (2) de la siguiente manera:

$$Cy + v = x \quad (6)$$

Esta corrupción exigirá una mejor constante RIC o un mayor número de medidas m para recuperar la señal.

Para alcanzar el objetivo planteado por CS, se han desarrollado diferentes algoritmos que se pueden clasificar en diferentes familias según sus prestaciones. Los dos parámetros característicos que varían de unos a otros son la velocidad, que influye en la latencia del sistema, y su capacidad de corrección contra errores. En la [Figura 5] podemos observar la clasificación de los algoritmos de CS:

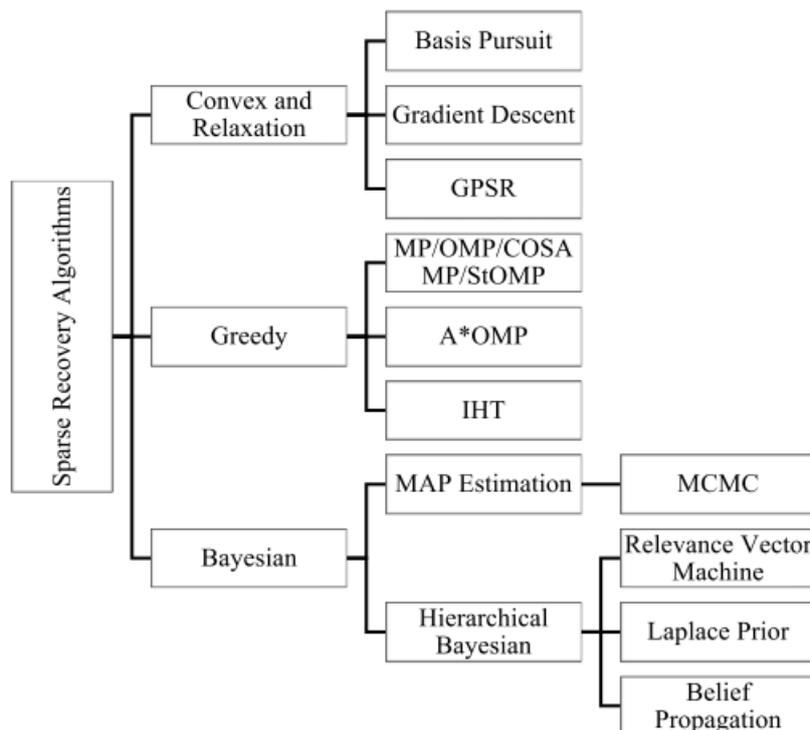


Figura 5. Clasificación de los algoritmos CS [15].

La primera familia, **Convex and Relaxation**, se basan en la idea de que el vector y correcto será el más disperso:

$$\text{mín } \|y\|_0; \quad Cy = x$$

Donde $\|\cdot\|_0$ indica la norma l_0 , el número de valores no nulos del vector. Al suponer una gran exigencia computacional, los algoritmos lo relajan al problema

$$\text{mín } \|y\|_1; \quad Cy = x$$

Donde $\|\cdot\|_1$ es la norma l_1 , la suma de los valores del vector. El algoritmo base de esta familia es el BP (*Basis Pursuit*) [16]. Esta familia se considera la mejor en cuanto a potencial capacidad para corregir errores, pero por otro lado suelen ser algoritmos muy lentos [15]. En el contexto de este trabajo, no parece la solución más adecuada, ya que se desconoce la cantidad de valores no nulos del vector disperso y además, para este caso se requiere una codificación y decodificación rápida (baja latencia), lo cual no sería posible empleando esta familia de algoritmos.

Por otro lado, existen los algoritmos pertenecientes a la familia **Bayesian**, que se basan en el *Machine Learning*, y exigen cierto conocimiento previo sobre el vector recibido x [17]. Al ser el vector aleatorio no hay ninguna forma de determinar con anterioridad cuál será el mensaje recibido. Estos algoritmos no estudian el problema de recuperación como un problema binario, si no que lo estudian como un problema probabilístico, que probabilidad hay de que los valores del vector k -disperso sean nulos o no. Por tanto, estos algoritmos permiten estimar con qué probabilidad el mensaje decodificado es correcto, lo cuál no es útil para nuestro caso. Además,

los algoritmos que más capacidad de corrección proporcionan exigen mucha capacidad de cálculo, que se traduce en términos de velocidad a un algoritmo más lento. Debido a la complejidad involucrada, se descartan los algoritmos de esta familia como posible solución al problema.

Por último, los algoritmos de la familia **Greedy** se basan en generar las combinaciones de k columnas de la matriz que generen el menor error posible cuando se comparen con el vector medido. De esta manera, se pueden conocer las posiciones de los valores no nulos del vector disperso y . La gran ventaja de estos algoritmos es que son muy rápidos comparados con las otras familias, cosa que es de vital importancia para nuestro caso. A pesar de que muchos algoritmos de esta familia tienen una capacidad de corrección contra errores muy pobre, con la introducción de MMP *Multipath Matching Pursuit*, su capacidad de corrección ha mejorado mucho [18].

Además del MMP, existen multitud de algoritmos en la bibliografía como el OMP *Orthogonal Matching Pursuit* o el CoSaMP *Compressive Sampling Matching Pursuit*, pero estos no mejoran la capacidad hallar las posiciones no nulas del vector disperso. Esto se debe a que el algoritmo MMP realiza una búsqueda múltiple, examinando varios candidatos a la vez. Por ello, este algoritmo es más lento que los mencionados anteriormente, pero mejora sus resultados en cuanto a capacidad de corrección, uno de los grandes problemas de esta familia.

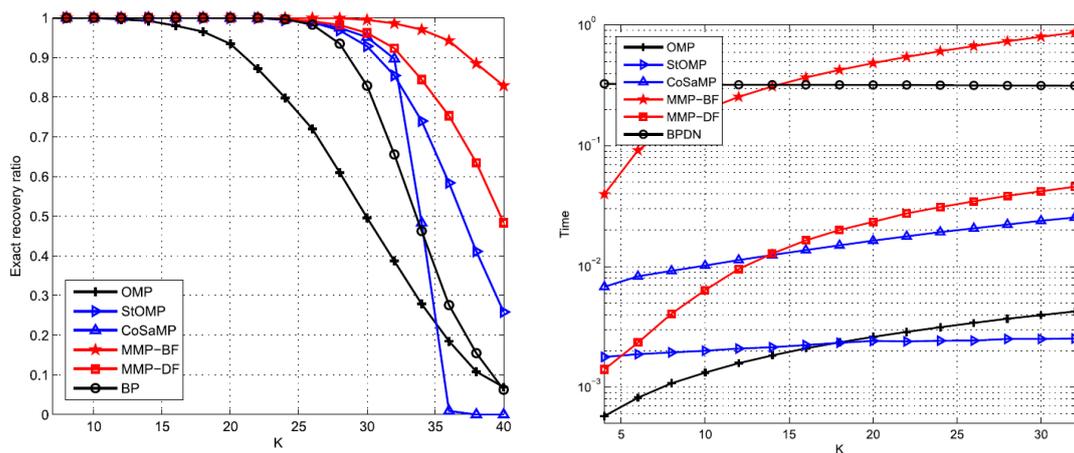


Figura 6. Rendimiento y velocidad de los algoritmos Greedy [18].

En la [Figura 6] se observa como en base a un estudio empírico realizado en [18] los algoritmos MMP son los más efectivos en cuanto a la recuperación exacta del mensaje enviado, es decir, en su capacidad de corrección de errores. Además, la nueva variante de este algoritmo que se presenta en este trabajo, el MMP-DF o *Depth First*, obtiene un rendimiento temporal comparable para valores más bajos de k .

En definitiva, los algoritmos de esta última familia parecen ser los más adecuados para el problema de los SVC, ya que además de que, si utilizan la información sobre el número de vectores no nulos que en este caso se posee, son muy interesantes debido a su velocidad, para así poder reducir la latencia del sistema.

Algoritmos CS	Métricas	
	Capacidad de recuperación	Velocidad
Basis Pursuit	Baja	Lento
Gradient Descent	Alta	Rápido
Orthogonal Matching Pursuit	Baja	Rápido
Multipath Matching Pursuit	Alta	Rápido
Iterative Hard Thresholding	Alta	Lento
Bayesian via Fast Laplace	Baja	Lento
Bayesian via Relevance Vector	Baja	Rápido

Tabla 1. Rendimiento y velocidad de los algoritmos CS [15].

6. SVC en las comunicaciones espaciales

Desde el comienzo de la era de la exploración espacial hasta la actualidad, se han enviado misiones al espacio para explorar nuestro Sistema Solar y más allá. Estas misiones, nos han permitido tener una mejor comprensión de nuestro planeta, de nuestro Sistema Solar y de nuestro Universo. Un factor clave para el éxito de cualquier misión son las comunicaciones, ya que las comunicaciones inalámbricas son la única forma de mantener un contacto entre la nave que se encuentra en el espacio y la Tierra. De esta manera, se puede monitorizar el estado de la nave desde la Tierra, transmitir comandos, recibir datos científicos sobre la misión o rastrear su posición o trayectoria.

La estructura de un sistema de comunicaciones espacial puede dividirse en dos segmentos principalmente:

- **Segmento espacial:** el segmento espacial incluye el satélite (o satélites) en órbita en el sistema, y la estación terrena que proporciona el control operativo de los satélites en órbita. La estación terrena se denomina de diversas formas: *Tracking, Telemetry, Command* (TT&C) o *Tracking, Telemetry, Command and Monitoring* (TTC&M). La estación TTC&M proporciona funciones esenciales de gestión y control para mantener el satélite en funcionamiento con seguridad en órbita [19].

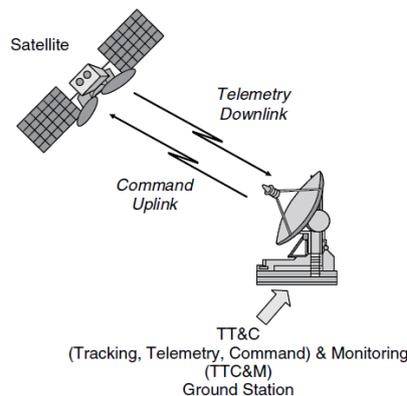


Figura 7. El segmento espacial para una red de satélites de comunicaciones [19].

- **Segmento terrestre:** el segmento terrestre del sistema de comunicaciones por satélite consta de todos los terminales terrestres que utilizan las capacidades de comunicaciones del segmento espacial. Las estaciones terrenas de TTC&M no están incluidas en el segmento terrestre [19].

Como se muestra en la [Figura 7], estos enlaces entre las estaciones terrenas y las naves pueden en algunos casos llegar a los billones de kilómetros para misiones que se encuentran al borde de nuestro Sistema Solar, como la misión *Voyager*, lanzada en 1977, y que sigue enviando mensajes a la Tierra. La capacidad de enviar señales de radiofrecuencia desde las naves espaciales es muy limitada en potencia, debido al incremento de peso que supone usar más

capacidad de hardware en el subsistema de comunicaciones, y las dificultades de diseño que conlleva. Es por ello que se intenta contrarrestar esta limitación, mediante el uso de grandes estaciones terrenas con una gran capacidad en potencia para enviar señales y con una gran capacidad de recepción de señales con potencias muy bajas.

En este contexto, la *Deep Space Network* (DSN) [20] es una red internacional de antenas utilizadas para las comunicaciones espaciales en la exploración del universo, para observaciones de radioastronomía y para apoyar misiones en órbita terrestre. La red, operativa desde 1963, ha sido de vital importancia desde la NASA para las comunicaciones con el espacio, realizando el seguimiento de múltiples misiones. La DSN consta de tres complejos de antenas, el primero de ellos ubicado en Australia a 40 km al sudoeste de Canberra, cerca de la reserva natural de Tidbinbilla. El segundo complejo, llamado Goldstone, se encuentra en Estados Unidos en el desierto de Mojave a 70 km al noroeste de la ciudad de Barstow (California), y el tercer complejo se encuentra en España, a 65 km al oeste de Madrid, y pertenece al término municipal de Robledo de Chavela. Estos complejos se encuentran a 120 o entre sí en la superficie terrestre, lo que permite que siempre haya visión directa con alguna de las antenas desde cualquier lugar del espacio. Además, aunque anteriormente y durante la mayor parte de su historia, cada complejo trabajaba de forma individual, en la actualidad mediante el protocolo “*Follow the Sun*”, los tres complejos trabajan de forma sincronizada, es decir, cada complejo se turna para ejecutar toda la red durante su turno de día y luego, al final del día, cede el control al siguiente complejo. El organismo que se encarga de administrar los complejos de la DSN es el *Jet Propulsion Laboratory* (JPL) de Pasadena, California. Las antenas de la DSN son generalmente reflectores parabólicos *Cassegrain* (reflector principal parabólico y sub-reflector hiperbólico) de gran tamaño con equipos a muy baja temperatura para lograr ganancias elevadas con bajo ruido, permitiendo así comunicaciones de muy largo alcance. Son usadas principalmente para rastrear, enviar comandos y recibir datos científicos de misiones con naves espaciales lejanas.

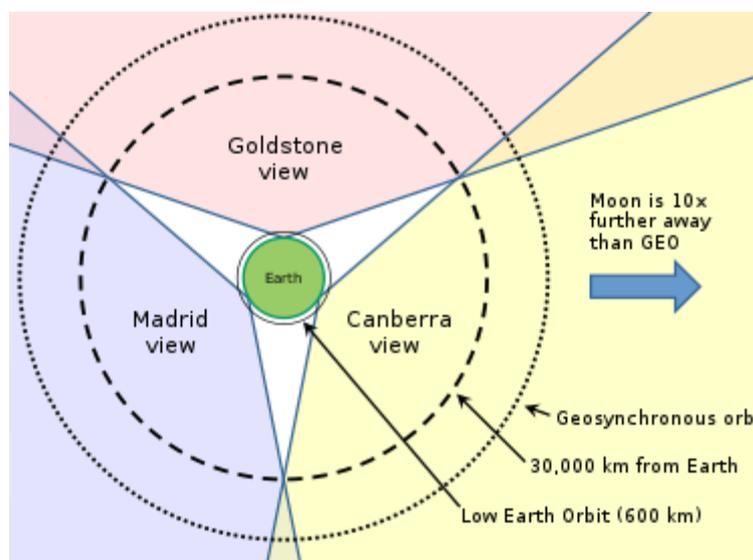


Figura 8. Deep Space Network (DSN), NASA.

A pesar de la ayuda que supone el contar con este tipo de infraestructuras para recibir y enviar señales a distancias tan lejanas, para poder realizar una comunicación optimizada y robusta (un requisito en muchos casos) es necesario hacer uso de la codificación contra errores. En este sentido, se han desarrollado códigos que se acercan al máximo teórico definido por *Shannon* [9] como, por ejemplo, una familia de códigos LDPC para transmisiones de alta tasa binaria. Sin embargo, para las comunicaciones con menor tasa binaria, estos y otros muchos códigos no son eficientes. Para comunicaciones de baja tasa binaria se han desarrollado otros códigos como los turbo-códigos, los polares, los BCH u otras familias de códigos LDPC que son más óptimos para estas condiciones. En el sector espacial, las misiones hacen uso de códigos antiguos [Tabla 1] que ofrecen peores rendimientos de los que podrían alcanzarse haciendo uso de los códigos más modernos que se vienen desarrollando. Es por ello que, en este trabajo, se proponen los SVC como una futura alternativa para la codificación contra errores en misiones espaciales.

Código	Misión
Sin codificar	Explorer, Mariner
Convolucional	Pioneer, Venus
Reed-Muller	Viking
Golay	Voyager
Reed Solomon	Galileo, Cassini, Mars pathfinder
Turbo-códigos	Messenger, Stereo, MRO
LDPC	MSL

Tabla 2. Códigos de corrección contra errores usados en diferentes misiones [7].

Estándares de Envío de Datos

En cualquier misión, desde la nave espacial, que puede estar a millones de kilómetros de distancia, se envían una gran cantidad de datos a la Tierra y viceversa. Sin tener en cuenta el envío de datos científicos específicos de cada misión que suelen genera una alta tasa binaria, los tipos de datos que se envían a través a través del sistema de comunicaciones que puede suponer longitudes de mensaje cortas, se pueden agrupar de la siguiente forma:

- **Telemetría:** el propósito de la telemetría es adquirir, procesar, decodificar y distribuir los datos de telemetría. Estos datos están formados por datos científicos y técnicos y sin transmitidos por las naves espaciales. Los estándares vienen establecidos por la *Consultative Committee for Space Data Systems (CCSDS)*. En la [Tabla 3] se muestran los requisitos de E_b/N_0 para los tamaños de bloque transmitidos soportados por la DSN [21].

Tamaño Bloque	Codificación y margen del enlace
8920 bits	Convolucional concatenado con Reed Salomon; $@E_b/N_0 \geq 1.8 \text{ dB}$
8920 bits	Turbo-código (R=1/3); $@E_b/N_0 \geq 0.4 \text{ dB}$
1784 bits	Turbo-código (R=1/6); $@E_b/N_0 \geq 0.4 \text{ dB}$
1024 bits	LDPC (R=1/2); $@E_b/N_0 \geq 2.4 \text{ dB}$

Tabla 3. Codificación y margen del enlace para diferentes tamaños de bloque [21].

- Comando: el objetivo del sistema de comando es controlar las actividades que realiza la nave enviando instrucciones desde la Tierra. En este caso los *Command Link Transmission Units (CLTUs)*, que serán los bloques de mensaje a ser codificados y enviados serán de un tamaño de entre 16 bits y 32.752 bits como máximo, con data rates de entre 7.8 bps a 20 Mbps. En cuanto a la codificación que se emplea, para el estándar de comandos de la DSN, se emplea Reed Solomon y diversos tipos de codificación LDPC (1/2, 2/3, 4/5, 7/8) [21].
- Tracking radiométrico: el tracking radiométrico permite calcular la posición y la velocidad de la nave en el espacio, para poder facilitar la comunicación con la DSN (apuntamiento).

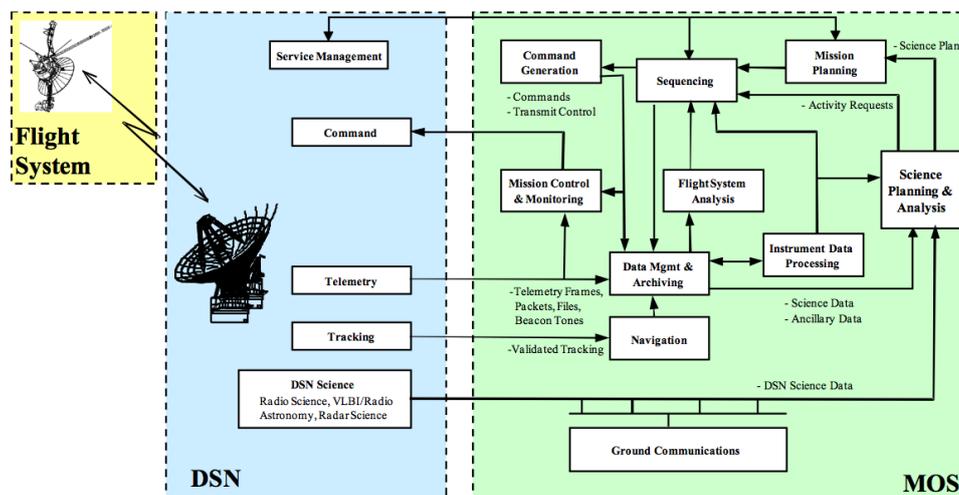


Figura 9. Contexto operaciones de la DSN [21].

7. Algoritmo de decodificación

Una vez enviado el mensaje por el canal, será la función del decodificador la de obtener el mensaje original que se ha enviado a partir de un mensaje recibido que posiblemente haya sido corrompido por el ruido. Este problema se basa en el planteado por CS, y por tanto el algoritmo de decodificación juega un papel clave a la hora de optimizar el sistema en cuanto a capacidad de corrección y velocidad (o latencia). Habitualmente se establecerá una relación entre estos dos parámetros con los que se jugará para obtener el resultado más óptimo posible. En este caso se hará uso del algoritmo *Multipath Matching Pursuit* (MMP) [18], un algoritmo perteneciente a la familia *Greedy*. Como se mencionó en el apartado anterior, esta familia de algoritmos que se pueden aplicar para el problema de CS es la más ideal para el caso de los SVC, ya que hacen uso de la información sobre el número de vectores no nulos que en este caso se posee y además se considera que son rápidos (reducen la latencia del sistema) en comparación con otros algoritmos. La idea principal detrás de este algoritmo es seleccionar un cierto número de vectores de codificación similares al vector ruidoso recibido. Una vez elegidos, se realizarán combinaciones lineales con el resto de los vectores de codificación para cada vector escogido en el paso anterior. Del resultado de estas combinaciones se escogen aquellos resultados que son más similares al vector ruidoso recibido. El proceso se repite un número de veces igual al número de k valores de nulos del vector k -disperso:

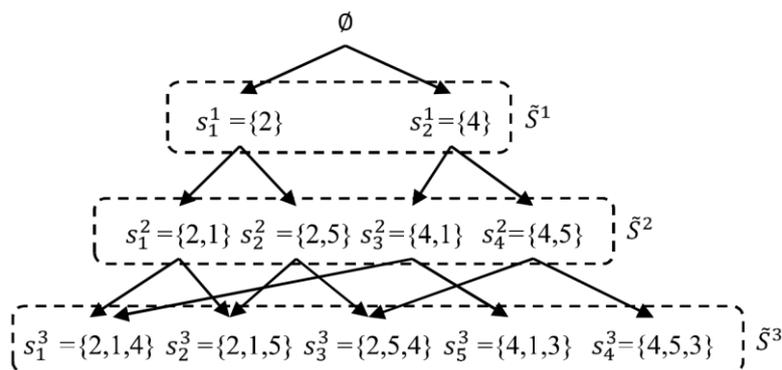


Figura 10. Algoritmo de búsqueda MMP.

En este caso se observa como se eligen en primer lugar, los dos vectores que más se parecen al vector ruidoso recibido. En segundo lugar, se han realizado combinaciones lineales de los dos vectores elegidos con el resto de los vectores hasta obtener las cuatro combinaciones que más se parecen al vector ruidoso. En el tercer nivel se repite la misma acción combinando las cuatro combinaciones obtenidas en el segundo nivel obteniendo ocho posibles combinaciones. El vector elegido será la combinación de tres vectores de código que más se parezca al vector ruidoso.

El fundamento matemático de este algoritmo es el siguiente, teniendo el vector ruidoso f y las palabras de código g_γ [18]:

$$f \in \mathcal{H}; g_\gamma \in \mathcal{D} \quad (7)$$

Siendo \mathcal{H} y \mathcal{D} subespacios de la misma dimensión. \mathcal{D} representa el diccionario que contiene palabras de código. Para representar f como la combinación lineal de un número k de palabras de código g_γ , se debe proyectar el vector f sobre un vector g_γ :

$$f = \langle f g_0 \rangle g_0 + Rf \quad (8)$$

Donde $\langle \rangle$ indica la proyección de un vector sobre otro. El objetivo es escoger g_0 como el vector que minimice el residuo Rf . Así se elige el vector más similar a f . Posteriormente, se procederá a realizar el mismo proceso con el resto:

$$Rf = \langle f g_1 \rangle g_1 + R^1 f \quad (9)$$

Al igual que en el caso anterior, g_1 será el vector que minimice el residuo $R^1 f$. Este proceso se realizará k veces, obteniendo una combinación lineal de k vectores, una aproximación al vector f . El MMP, realiza este mismo método, pero en cada paso selecciona más de un vector, para que la probabilidad de encontrar la solución que minimice el último residuo R^{k-1} sea mayor.

El algoritmo que se ha implementado en la plataforma de simulación viene descrito por el siguiente pseudocódigo [18]:

THE MMP ALGORITHM

Input: measurement \mathbf{y} , sensing matrix Φ , sparsity K , number of path L
Output: estimated signal $\hat{\mathbf{x}}$
Initialization: $k := 0$ (iteration index), $\mathbf{r}^0 := \mathbf{y}$ (initial residual), $S^0 := \{\emptyset\}$

while $k < K$ **do**
 $k := k + 1$, $u := 0$, $S^k := \emptyset$
 for $i = 1$ **to** $|S^{k-1}|$ **do**
 $\tilde{\pi} := \arg \max_{|\pi|=L} \|(\Phi' \mathbf{r}_i^{k-1})_\pi\|_2^2$ *(choose L best indices)*
 for $j = 1$ **to** L **do**
 $s_{temp} := s_i^{k-1} \cup \{\tilde{\pi}_j\}$ *(construct a temporary path)*
 if $s_{temp} \notin S^k$ **then** *(check if the path already exists)*
 $u := u + 1$ *(candidate index update)*
 $s_u^k := s_{temp}$ *(path update)*
 $S^k := S^k \cup \{s_u^k\}$ *(update the set of path)*
 $\hat{\mathbf{x}}_u^k := \Phi_{s_u^k}^\dagger \mathbf{y}$ *(perform estimation)*
 $\mathbf{r}_u^k := \mathbf{y} - \Phi_{s_u^k} \hat{\mathbf{x}}_u^k$ *(residual update)*
 end if
 end for
 end for
 end while
 $u^* := \arg \min \| \mathbf{r}_u^K \|_2^2$ *(find index of the best candidate)*
 $s^* := s_{u^*}^K$
 return $\hat{\mathbf{x}} = \Phi_{s^*}^\dagger \mathbf{y}$

A partir de ahora se referirá al algoritmo descrito como MMP *Breadth-First* o MMP-BF. En la bibliografía se propone una modificación del MMP-BF, denominada MMP *Depth-First* (MMP-DF), que permite controlar la complejidad computacional de MMP obteniendo un rendimiento similar.

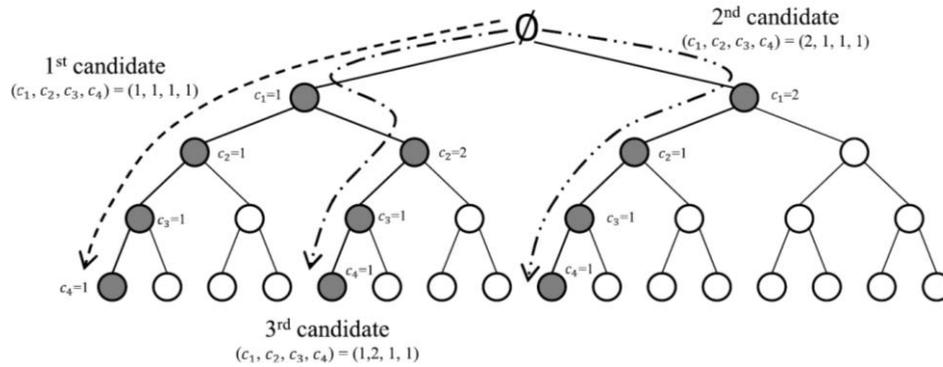


Figura 11. Algoritmo de búsqueda MMP-DF.

El algoritmo MMP-BF realiza una investigación paralela de candidatos, examinando múltiples candidatos prometedores y combinando un gran número de candidatos en el medio de la búsqueda, lo que supone una mayor complejidad de cálculo. El algoritmo MMP-DF sin embargo, realiza la búsqueda de manera secuencial, y se finaliza si la magnitud del residuo satisface una condición de terminación adecuada o si el número de candidatos alcanza un valor máximo predefinido N_{max} . Estos parámetros permiten controlar y limitar la complejidad del algoritmo MMP-DF, lo que supone un ahorro significativo en el costo computacional en comparación con el algoritmo MMP-BF.

Se ha implementado también el algoritmo MMP-DF en la plataforma de simulación para poder realizar un análisis comparativo (se muestra en el siguiente apartado). Viene descrito por el siguiente pseudocódigo [18]:

THE MMP-DF ALGORITHM

Input:
 Measurement \mathbf{y} , sensing matrix Φ , sparsity K , number of expansion L ,
 stop threshold ϵ , max number of search candidate ℓ_{\max}

Output:
 Estimated signal $\hat{\mathbf{x}}$

Initialization:
 $\ell := 0$ (candidate order), $\rho := \infty$ (min. magnitude of residual)

while $\ell < \ell_{\max}$ and $\epsilon < \rho$ **do**
 $\ell := \ell + 1$
 $\mathbf{r}^0 := \mathbf{y}$
 $[c_1, \dots, c_K] := \text{compute_ck}(\ell, L)$ *(compute layer order)*
for $k = 1$ **to** K **do** *(investigate ℓ -th candidate)*
 $\tilde{\pi} := \arg \max_{|\pi|=L} \|(\Phi' \mathbf{r}^{k-1})_{\pi}\|_2^2$ *(choose L best indices)*
 $s_{\ell}^k := s_{\ell}^{k-1} \cup \{\tilde{\pi}_{c_k}\}$ *(construct a path in k -th layer)*
 $\hat{\mathbf{x}}^k := \Phi_{s_{\ell}^k}^{\dagger} \mathbf{y}$ *(estimate $\hat{\mathbf{x}}^k$ in k -th layer)*
 $\mathbf{r}^k := \mathbf{y} - \Phi_{s_{\ell}^k} \hat{\mathbf{x}}^k$ *(update residual)*
end for
if $|\mathbf{r}^K| < \rho$ **then** *(update the smallest residual)*
 $\rho := |\mathbf{r}^K|$
 $\hat{\mathbf{x}}^* := \hat{\mathbf{x}}^K$
end if
end while
return $\hat{\mathbf{x}}^*$

function $\text{compute_ck}(\ell, L)$
 $temp := \ell - 1$
for $k = 1$ **to** K **do**
 $c_k := \text{mod}(temp, L) + 1$
 $temp := \text{floor}(temp/L)$
end for
return $[c_1, \dots, c_K]$
end function

Rendimiento

Antes de plantear la simulación del sistema de comunicaciones, cabe destacar la relación entre el número de k valores no nulos del vector k -disperso y la modulación. El número de valores no nulos cambia la constelación de modulación. A la hora de realizar la simulación con plataforma software se ha escogido un valor de $k = 2$, que se corresponde con la modulación QPSK, ya que, en el contexto de aplicabilidad de los SVC, estos son óptimos para comunicaciones robustas.

Se ha adaptado la plataforma de simulación empleada por Iñigo Bilbao [7] en Matlab, para el uso del nuevo algoritmo. En primer lugar, usando los mismos parámetros de entrada [Tabla 4],

se ha procedido a obtener la misma “curva de BLER” que la expuesta en la bibliografía. La [Figura 12] muestra los resultados obtenidos por Hyoung-ju et al. El eje horizontal representa los valores de SNR. Es importante hacer notar que el sistema propuesto por Hyoung-ju es una combinación de codificación y redundancia basada en multiplexación por división en frecuencia. El mismo paquete se transmite de forma simultánea en 8 portadoras. Por lo tanto, para poder comparar los resultados es necesario hacer una normalización la relación SNR, añadiendo el factor $10 \log_{10}(8)$ que resulta en aproximadamente 9 dB. Es decir, en la [Figura x] el eje X debería estar desplazado 9 unidades hacia la izquierda. Así, la curva indica un $BLER = 10^{-5}$ para un valor de SNR de aproximadamente 3 dB.

Parámetro	Valor
Longitud de mensaje original	$b = 12 \text{ bits}$
Matriz de codificación (C)	Matriz de Bernoulli
Valores no nulos del vector k-disperso	$k = 2$
Code Rate	$R = \frac{1}{7}$

Tabla 4. Parámetros sistema comunicaciones

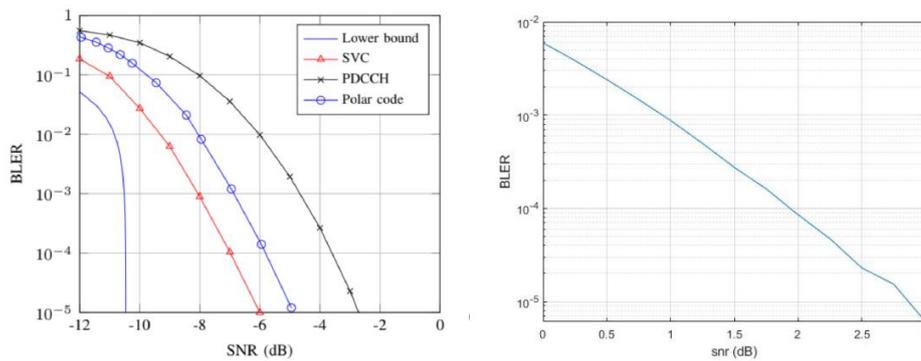


Figura 12. Validación sistema bibliografía [5].

Usando los mismos datos y modificando el *Code Rate*, además de eligiendo un número máximo de candidatos de tres y un *stop threshold* de uno, se ha realizado la simulación de un sistema de comunicaciones empleando ambos algoritmos de decodificación, obteniendo el siguiente resultado:

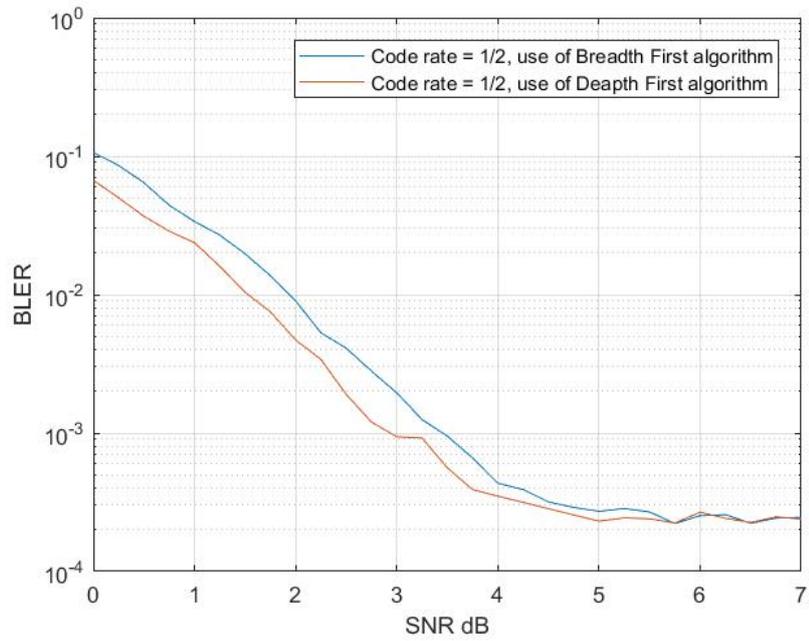


Figura 13. Curva BLER para $R=1/2$.

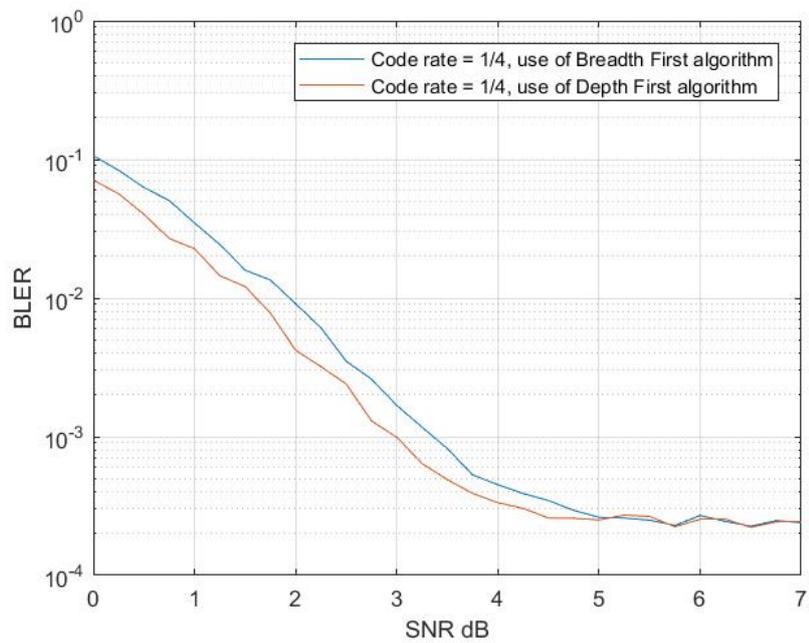


Figura 14. Curva BLER para $R=1/4$

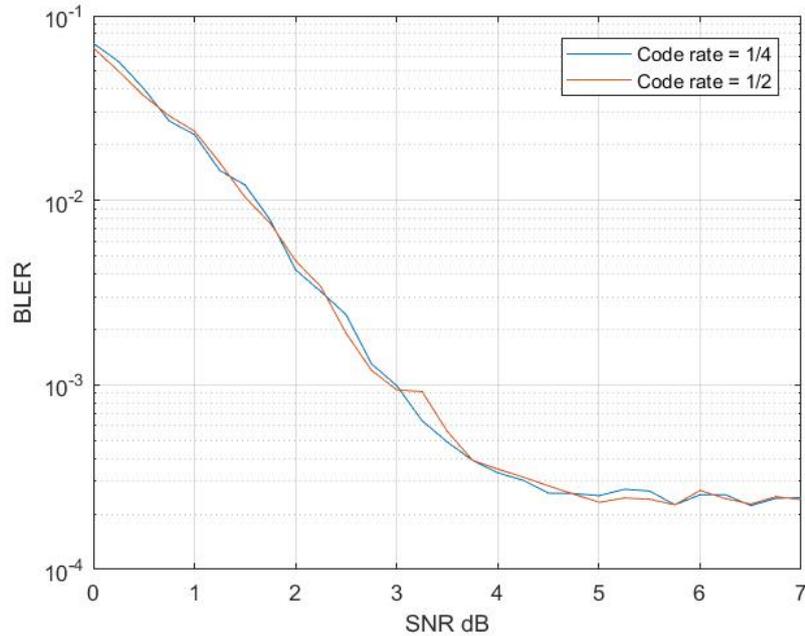


Figura 15. Curva BLER MMP-DF, para diferentes R .

A partir de las curvas anteriores se puede deducir que ambos algoritmos tienen un rendimiento muy similar en cuanto a Block Error Rate (BLER) para el rango de valores de SNR de 0-7 dB. Incluso se puede distinguir como el nuevo algoritmo propuesto, el algoritmo MMP-DF, tiene un rendimiento ligeramente superior en cuanto a capacidad de corrección de errores.

Complejidad

Una vez verificado que ambos algoritmos tienen un rendimiento similar en cuanto a corrección de errores, conviene realizar un estudio acerca de su complejidad computacional. Caracterizarla es de vital importancia, debido a su relación directa con la velocidad y la latencia del sistema. Para poder analizar la complejidad de un algoritmo, se puede emplear la notación Big O, una forma matemática de expresar de forma sencilla cuanto tarda un algoritmo en ejecutarse, y así poder realizar un estudio comparativo con otros. En concreto se usa para expresar de forma abstracta la velocidad de procesamiento de un algoritmo, atendiendo a cómo aumenta su tiempo de ejecución en función de un aumento significativo del tamaño de los datos de entrada [23]. En este sentido, si se toma una aproximación a partir de la función Big O del algoritmo OMP (*Orthogonal Matching Pursuit*) descrito en [13] [24], se puede obtener la función Big O para MMP:

$$O\left(\frac{7}{2}b \log \frac{7}{2}b\right) \quad (10)$$

En este caso, la complejidad descrita por la función Big O será la misma para ambos algoritmos, MMP-BF y MMP-DF, debido a la estructura y parámetros de entrada de ambos.

A pesar de ello, se ha logrado encontrar un método alternativo para poder medir y estimar la complejidad de un algoritmo, y facilitar así su estudio comparativo. El método utilizado serán las “Medidas de Complejidad de Halstead” (*Halstead complexity measures*) [25]. Este método se basa en métricas de software introducidas por Maurice Howard Halstead en 1977. Halstead sugirió que las métricas del software deberían reflejar la implementación o expresión de algoritmos en diferentes lenguajes, pero ser independientes de su ejecución en una plataforma específica. Por lo tanto, estas métricas se calculan estáticamente a partir del código del programa. El objetivo de Halstead era identificar las propiedades medibles del software y las relaciones entre ellas. Estas propiedades medibles se basan en las métricas sobre cantidad de operadores y operandos diferentes que se encuentran en el programa. Las métricas que sugirió realizar para calcular la complejidad son las siguientes:

- n_1 : número de operadores únicos.
- n_2 : número de operandos únicos.
- N_1 : número total de ocurrencias de operadores.
- N_2 : número total de ocurrencias de operandos.
- $n = n_1 + n_2$
- $N = N_1 + N_2$

La primera métrica a tener en cuenta es el volumen de programa V, que es proporcional al tamaño del programa, y representa el tamaño, en bits, del espacio necesario para almacenar el programa. Este parámetro depende de la implementación específica del algoritmo:

$$V = N \log_2 n \quad (11)$$

La segunda métrica es el nivel de programa L (program level), que indica la dificultad para la comprensión del programa. Tiene que ver con el nivel de abstracción del lenguaje de programación empleado. Varía entre cero y uno, siendo uno el nivel más alto posible:

$$L = \frac{2 n_2}{n_1 N_2} \quad (12)$$

La siguiente métrica tiene que ver con aspectos psicológicos de la complejidad. Halstead estudió los procesos cognitivos relacionados con la percepción y retención de estímulos simples. El número medio de discriminaciones mentales por segundo en un ser humano promedio, también llamado número de *Stroud*, está entre 5 y 20. En su modelo, el número de discriminaciones realizadas en la elaboración de un programa, llamado esfuerzo E, es dada por:

$$E = \frac{V}{L} \quad (13)$$

La dificultad del programa D, describe la dificultad en el manejo del programa:

$$D = \frac{n_1 N_2}{2 n_2} = \frac{1}{L} \quad (14)$$

Por último, la complejidad ciclomática, mide el número de caminos linealmente independientes a través de un programa. Esta medida proporciona un solo número que se puede usar para comparar con la complejidad de otros programas. Se refiere a ella a menudo como dificultad de programa o complejidad de McCabes.

Realizando esta serie de medidas se puede obtener un buen análisis comparativo de ambos algoritmos para varios lenguajes de programación diferentes como los recogidos en [25]:

Algorithm Name	Language	Program Vol(V)	Program Difficulty(D)	Program Effort(E)	Cyclomatic No V(G)
Breadth-First Search	C	733	20	14660	5
Breadth-First Search	C++	998	17	16966	5
Breadth-First Search	Pascal	640	9	5760	3
Breadth-First Search	Visual BASIC	1197	10	11970	6

Tabla 5. Medidas de complejidad del algoritmo Breadth-First para diferentes lenguajes de programación.

Algorithm Name	Language	Program Vol(V)	Program Difficulty(D)	Program Effort(E)	Cyclomatic No V(G)
Depth-First Search	C	491	16	7851	5
Depth-First Search	C++	515	18	9270	5
Depth-First Search	Pascal	539	8	4312	5
Depth-First Search	Visual BASIC	1069	9	9627	6

Tabla 6. Medidas de complejidad del algoritmo Depth-First para diferentes lenguajes de programación.

Las medidas de las tablas x y x nos sirven como una buena referencia y aproximación para evaluar desde un punto de vista comparativo los algoritmos MMP-BF y MMP-DF, ya que solo varían en el método de búsqueda de candidatos. Se observa como hay una dificultad notable en cuanto a volumen de programa, que puede relacionarse con los recursos de memoria necesarios, además de el esfuerzo del programa, que es superior en el caso de *Depth-First*. En cuanto a la dificultad de programa y la complejidad ciclomática, se obtienen resultados similares. Por tanto, se puede concluir que el algoritmo MMP-DF propuesto supone un ahorro significativo en complejidad sin sacrificar su capacidad de corrección de errores frente al MMP-BF.

Caso de uso

Por último, se busca validar la implementación de los SVCs para un sistema de comunicaciones espaciales, por lo que se ha buscado adaptar los mensajes a enviar a las características de los mensajes descritos por los estándares de telemetría en el [Capítulo 6]. Se pretendía realizar simulaciones usando el tamaño de mensaje empleado para la recepción de datos de telemetría por la DSN, pero debido a las limitaciones en la capacidad de cálculo, ha sido imposible desarrollar un sistema para el tamaño de mensaje mínimo de $b = 1024 \text{ bits}$. Por tanto, se ha decidido realizar una simulación de los datos de comando enviados por la DSN hacia las naves

espaciales, que tienen una longitud menor, en este caso de $b = 16 \text{ bits}$ (longitud mínima). Se han obtenido las siguientes curvas de BLER para diferentes valores de SNR:

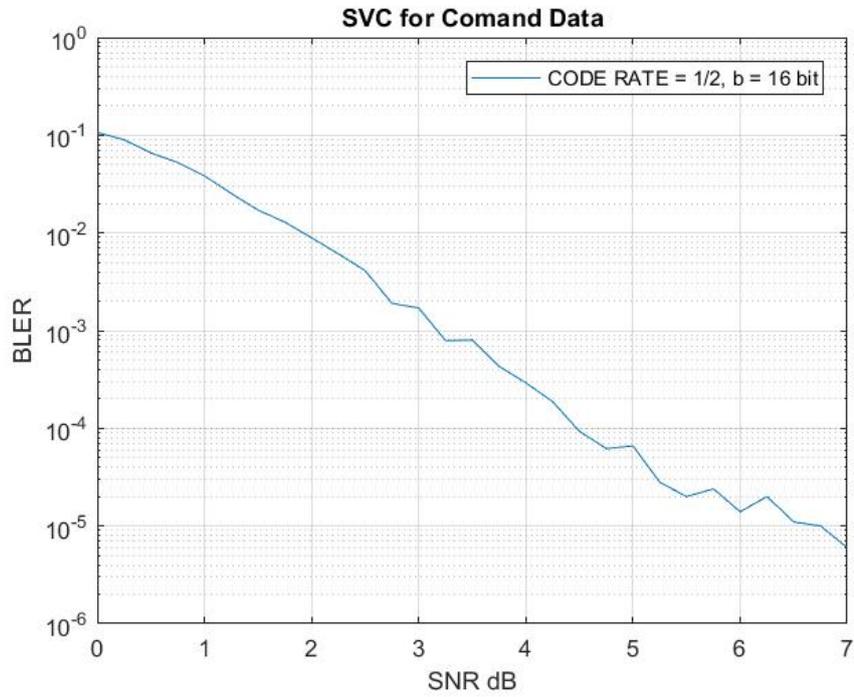


Figura 16. Curva BLER para envío de comandos.

Se ve como la curva obtenida, para valores bajos de SNR, tiene un comportamiento similar al de las simulaciones anteriores (12 bits). Esto implica la validación y posible futura implementación de esta novedosa técnica de codificación contra errores para el sistema de comunicaciones de misiones espaciales.

8. Conclusiones

Mediante el presente Trabajo Fin de Máster, se ha contribuido al diseño de una arquitectura de sistema de comunicaciones inalámbricas para el envío de mensajes cortos utilizando los códigos SVC (*Sparse Vector Codes*) como sistema de protección contra errores. Se ha buscado de forma específica, un algoritmo de decodificación en el receptor que mejorara o ampliara el rendimiento de los utilizados anteriormente.

Se ha comenzado por un estudio del estado del arte de los códigos de corrección de errores para mensajes cortos SVC para poder tener una comprensión general acerca de esta técnica de codificación. Se ha realizado un análisis desde el ámbito de las comunicaciones espaciales, ya que dentro de este marco es característico el envío de mensajes cortos (telemetría, telecomando, etc...), y se podría optimizar de forma sustancial el envío de información ya que los códigos empleados actualmente están optimizados para mensajes largos. Se ha realizado también un estudio acerca de los estándares actuales en materia de transmisión de datos en la DSN (*Deep Space Network*), que es la red de antenas que constituye la estación terrena encargada de llevar a cabo las comunicaciones con las misiones.

Una vez planteado el estado del arte, se ha diseñado una plataforma de simulación para un sistema de comunicaciones usando los códigos SVC y empleando una nueva variante del algoritmo presentado en anteriores estudios. Este nuevo algoritmo, MMP-DF (*Multipath Matching Pursuit- Depth First*), controla la complejidad del algoritmo usado anteriormente mediante el uso de dos parámetros adicionales, una condición de terminación y el número máximo de candidatos. Los resultados en cuanto a rendimiento muestran una leve mejoría respecto al anterior algoritmo. Respecto a la complejidad, mediante las “Medidas de Complejidad de Halstead” se ha podido realizar un estudio comparativo concluyente. El algoritmo MMP-DF supone un gran ahorro de recursos de memoria debido al menor volumen de programa respecto al anterior MMP-BF, por lo que supondrá una menor latencia para el sistema de comunicaciones, y una notable mejoría respecto al diseño anterior. Además, se ha realizado una simulación con un caso de uso en el que se envían datos de comando desde la DSN con un mensaje de 16 bits, obteniendo un resultado parecido al obtenido en los apartados anteriores.

Es por ello, y tras un estudio de un algoritmo de decodificación óptimo, que se plantean estos nuevos códigos como posibles candidatos para futuras misiones espaciales, ya que los códigos empleados en la actualidad no son igual de eficientes para longitudes de mensaje cortas.

Finalmente, a la vista de los resultados obtenidos en este Trabajo Fin de Máster cabe considerar los siguientes desarrollos futuros y posibles mejoras:

- Realizar un estudio detallado de los parámetros que controlan la complejidad del algoritmo MMP-DF.
- Optimizar el uso de estos códigos mejorando la matriz de codificación. Usar un método alternativo al planteado en estudios anteriores para generar columnas menos correladas entre sí.
- Realizar simulaciones con hardware que permita una mayor capacidad de cálculo para ajustar las curvas de BLER para SNRs altos.

9. Bibliografía

- [1] Molisch, A. F. (2012). *Wireless communications*. John Wiley & Sons.
- [2] Insider, T. (2018, 9 abril). Here's how much money it actually costs to launch stuff into space. *Business Insider*. <https://www.businessinsider.com/spacex-rocket-cargo-price-by-weight-2016-6?IR=T>
- [3] LAURETTA, D. S., et al. OSIRIS-REx: sample return from asteroid (101955) Bennu. *Space Science Reviews*, 2017, vol. 212, p. 925-984.
- [4] General Dynamics. (2019). *Small Deep-Space Transponder (SDST)*. Mission Systems, General Dynamics.
- [5] H. Ji, S. Kim y B. Shim, "Sparse vector coding for ultra short packet transmission", *2018 Information Theory and Applications Workshop, ITA 2018*, nº 2, págs. 1-9, 2018. DOI: 10.1109/ITA.2018.8503179.
- [6] D. L. Donoho, "Compressed sensing", *IEEE Transactions on Information Theory*, vol. 52, nº 4, págs. 1289-1306, 2006, ISSN: 00189448. DOI: 10.1109/TIT.2006.871582.
- [7] Bilbao, I. (2020). *Sparse Vector Coding*. University Basque Country.
- [8] Objetivos de Desarrollo Sostenible. 17 objetivos para transformar nuestro mundo. Naciones Unidas. <https://www.un.org/sustainabledevelopment/es/>
- [9] C. E. Shannon, "A mathematical theory of communication", *The Bell System Technical Journal*, vol. 27, nº 3, págs. 379-423, 1948.
- [10] D. J. C. Shu Lin, Error control coding: Fundamentals and Applications. Prentice-Hall Computer Applications in Electrical Engineering Series, 1983, vol. 1, ISBN: 0-13-283796-X.ç
- [11] Hsu, C. H., & Anastasopoulos, A. (2008). Capacity achieving LDPC codes through puncturing. *IEEE Transactions on Information Theory*, 54(10), 4698-4706.
- [12] Shirvanimoghaddam, M., Mohammadi, M. S., Abbas, R., Minja, A., Yue, C., Matuz, B., ... & Vucetic, B. (2018). Short block-length codes for ultra-reliable low latency communications. *IEEE Communications Magazine*, 57(2), 130-137.
- [13] Bilbao, I., Fanari, L., Iradier, E., Angueira, P., & Montalban, J. (2022). Sparse Vector Coding for Short-Packet Transmission on Industrial Communications: Reference Architecture and Design Challenges. *IEEE Open Journal of the Industrial Electronics Society*, 4, 1-13.
- [14] E. J. Candes y M. B. Wakin, "An Introduction To Compressive Sampling", *IEEE Signal Processing Magazine*, vol. 25, nº 2, págs. 21-30, 2008.
- [15] Y. Arjoune, N. Kaabouch, H. El Ghazi y A. Tamtaoui, "Compressive sensing: Performance comparison of sparse recovery algorithms", *2017 IEEE 7th Annual Computing and*

Communication Workshop and Conference, CCWC 2017, págs. 1-7, 2017. DOI: 10.1109/CCWC.2017.7868430.

[16] D. A. Lorenz, M. E. Pfetsch y A. M. Tillmann, "Solving basis pursuit: Heuristic optimality check and solver comparison", *ACM Transactions on Mathematical Software*, vol. 41, nº 2, 2015, ISSN: 15577295. DOI: 10.1145/2689662.

[17] S. Ji, Y. Xue y L. Carin, "Bayesian Compressive Sensing", *IEEE Transactions on Signal Processing*, vol. 56, nº 6, págs. 2346-2356, 2008.

[18] Kwon, S., Wang, J., & Shim, B. (2014). Multipath matching pursuit. *IEEE Transactions on Information Theory*, 60(5), 2986-3001.

[19] Ippolito Jr, L. J. (2017). *Satellite communications systems engineering: atmospheric effects, satellite link design and system performance*. John Wiley & Sons.

[20] NASA. (2020). *About the Deep Space Network*: <https://deepspace.jpl.nasa.gov/about/>

[21] Network, D. S. (2022). *Deep Space Network Services Catalog* (No. 820-100). Tech. Rep. DSN.

[22] J. Wang. (2020). Multipath matching pursuit with breadth-first, dirección: <https://www.mathworks.com/matlabcentral/fileexchange/57994-multipath-matching-pursuit-with-breadth-first-mmp-bf>

[23] Cianes,P. (2020). Notación big O. *Programming Basics*: <https://pablocianes.com/notacion-big-o/>

[24] Zhang, R., Shim, B., Lou, Y., Jia, S., & Wu, W. (2020). Sparse vector coding aided ultra-reliable and low-latency communications in multi-user massive MIMO systems. *IEEE Transactions on Vehicular Technology*, 70(1), 1019-1024.

[25] Halstead, Maurice H. (1977). *Elements of Software Science*. Amsterdam: Elsevier North-Holland, Inc. ISBN 0-444-00205-7.