

4. CONCLUSIONES

Hemos realizado un estudio de los principales algoritmos de decodificación y podemos concluir que de las opciones tratadas, las más adecuadas son la aproximación lineal de la tanh (PW, piecewise) y la aproximación por sustitución de las exponenciales en el enfoque de Gallager por potencias en base 2 (FI). La aproximación lineal obtiene mejores resultados pero tiene una implementación más complicada y lenta. Si necesitamos alta precisión y no nos preocupa la velocidad o el área del circuito elegiremos la aproximación PW y si requerimos velocidad y una implementación simple, elegiremos la FI. El algoritmo LUT no ha obtenido muy buenos resultados pero puede mejorarse si se aumenta el tamaño de la LUT y el algoritmo MS también puede mejorarse si se aplican las correcciones que explicábamos en el capítulo correspondiente, que no suponen un gran aumento computacional. Es posible que con estas modificaciones, el algoritmo MS fuese una muy buena opción a tener en cuenta. Si lo que queremos es una implementación software del decodificador podemos optar por los algoritmos matemáticos ideales, teniendo en cuenta el efecto que tiene la precisión finita de los operandos en la evaluación de la tanh.

Algunas de las mejoras y posibles líneas de estudio para futuras ampliaciones de este proyecto podrían ser:

- Aumentar el número máximo de palabras enviadas para obtener resultados más precisos.
- Incluir otros algoritmos, por ejemplo basados en el enfoque jacobiano.
- Implementar los algoritmos en hardware mediante una FPGA y comprobar los resultados de las simulaciones, estudiar el efecto de la precisión finita, el área necesaria, consumo y velocidad de decodificación [17][15].
- Realizar un estudio de los turbocódigos y realizar una comparación entre ellos tanto para los algoritmos ideales como para las implementaciones.[27]
- Estudiar los algoritmos con otras modulaciones y canales y ver su efecto [12][30].
- Ampliar el análisis del algoritmo minsum, incluyendo las correcciones [10][4][18].
- Estudiar el efecto del número de iteraciones del algoritmo en la BER [16].
- Estudiar el efecto de la longitud del código.
- Estudiar también la codificación usando códigos LDPC.

- Además de usar simplificaciones para los nodos de chequeo, simplificar también los nodos de bit: Algoritmos APP (a posteriori probabilities) [8].