

Desarrollo de una interfaz gráfica en MATLAB[®] para simulaciones de códigos turbo

Johana Nohelia Cárdenas Cardoso, Alejandro Torres Herrera, Daniel Andrés Vásquez Cano

Resumen— Este artículo realiza una descripción del proceso realizado para el desarrollo de una interfaz gráfica de usuario en MATLAB[®] que soporta la simulación de códigos turbo, permitiendo la variación de sus características más representativas, y el desarrollo de gráficas comparativas, utilizando como estructura base un modelo de sistema de comunicación digital, implementado en el lenguaje de programación C++.

Palabras claves— Códigos turbo, MATLAB[®].

I. INTRODUCCIÓN

Los códigos turbo [1] son el avance más importante en los últimos tiempos, en el área de codificación de canal en sistemas de comunicación digital. Debido a la necesidad de su constante estudio para el conocimiento correcto y total de su funcionamiento, los investigadores hacen continuamente diferentes simulaciones, modificando todas las posibles variables que afectan su desempeño.

Estas simulaciones generalmente se hacen en formatos muy básicos, ya que son desarrolladas por las mismas personas que las analizan, y en general constan de pantallas sencillas que muestran los resultados numéricos de las simulaciones planteadas. Más adelante, si es necesario, se grafican estos datos por medio de cualquier herramienta que lo permita.

Surge entonces la idea de diseñar una herramienta de simulación para códigos turbo, que no sólo permita un acercamiento directo a este esquema de codificación, sino que además posea una interfaz para sus usuarios, que admita la comprensión y el análisis de resultados de forma clara e inmediata.

Este artículo presenta el desarrollo de una interfaz gráfica en MATLAB[®] para la simulación de códigos turbo, comenzando con una explicación sencilla del sistema de comunicación modelado, e implementado en lenguaje C++. Prosigue con una explicación del proceso que se realizó para adaptar el modelo en C++, a una función tal que MATLAB[®] estuviera en capacidad de utilizar como base para los cálculos de las simulaciones deseadas. Por último se da una descripción del desarrollo final de la interfaz, y se muestran resultados

gráficos obtenidos por medio de esta herramienta de simulación.

II. GENERALIDADES DEL SISTEMA DE COMUNICACIÓN MODELADO

Para el diseño del modelo que permitiera la simulación del esquema de codificación turbo se hizo el análisis de investigaciones y estudios previos, buscando implementar bloques básicos y óptimos, que al final obtuvieran resultados con datos y gráficas consistentes.

Los bloques que conforman el modelo realizado se observan en la Fig. 1. Estos elementos fueron desarrollados utilizando el programa Microsoft Visual C++ versión 6.0, aprovechando las características de eficiencia y rapidez de su compilador, ya que los cálculos que implican estas simulaciones así lo requieren. También se empleó una metodología de constante diseño, simulación, y optimización.

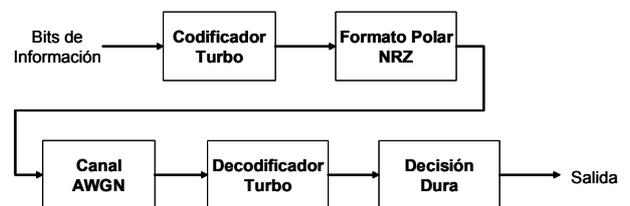


Fig. 1. Diagrama de bloques del modelo

Se observa en este diagrama de bloques que el modelo desarrollado va desde el bloque codificador de canal en el transmisor, hasta el bloque decodificador de canal en el receptor, de un sistema de comunicación digital.

La labor en conjunto de estos módulos es la base de la herramienta de simulación de códigos turbo. Las funciones en C++ que implementan este modelo tienen varios parámetros de entrada y de salida que ofrecen la posibilidad de realizar diferentes simulaciones de este esquema de codificación.

III. ADAPTACIÓN DEL MODELO PARA EL ENTORNO DE MATLAB[®]

Una vez se realizó el desarrollo de las funciones en C++ que implementaban los bloques del modelo planteado, y se comprobó su correcto funcionamiento, se procedió a realizar

la interfaz de usuario. Esta interfaz se implementó en MATLAB® versión 6.5, dada la familiaridad con las herramientas encontradas en el directorio *matlab\uitools* que facilitan la creación de interfaces con el usuario.

Debido a que la interfaz se creó en MATLAB®, el primer paso para que interactuara con las funciones ya realizadas en C++, fue adaptar dichas funciones para que MATLAB® las pudiera interpretar, y así, las pudiera utilizar como cualquier otra subrutina, en la aplicación de la interfaz. Para lograr esto, se utilizó la Interfaz de Programas de Aplicación de MATLAB® (API, por su siglas en inglés), donde MATLAB® permite trabajar con rutinas externas, escritas en otros lenguajes de programación como C, Fortran o Java [2].

Para llamar un programa creado en C++ desde MATLAB®, es necesario realizar una adaptación del programa básico en C++, y lograr así que sea posible el intercambio de datos entre MATLAB® y la función. Una vez se hace este ajuste, utilizando las diferentes funciones *C MX* y *C MEX* de MATLAB®, se efectúa una nueva compilación del programa, esta vez desde MATLAB® con el comando “*mex*”, y si dicha compilación finaliza correctamente, al final de la operación se genera un archivo con extensión *.dll* (si se está utilizando la plataforma Windows), del mismo nombre que la función original en C++. Este tipo de archivos que genera MATLAB® se conocen como archivos MEX.

Los archivos MEX son los que realmente trabajan con MATLAB®, funcionando como si fueran archivos con extensión *.m* típicos. Esta es una característica muy importante debido a que, aunque las funciones originales son escritas en C++ y necesitan un compilador de C++, los archivos MEX creados a partir de ellas sólo necesitan MATLAB® para trabajar.

IV. INTERFAZ GRÁFICA DE LA HERRAMIENTA DE SIMULACIÓN PARA CÓDIGOS TURBO

Una vez se tuvieron los archivos MEX respectivos, el resto del desarrollo de la interfaz se hizo en MATLAB®, utilizando la herramienta GUIDE, que permite crear interfaces de usuario de manera ágil y sencilla. GUIDE permite realizar conjuntos de pantallas, con botones, menús, ventanas, etc., generando dos archivos con el mismo nombre para cada pantalla creada: un archivo con extensión *.m* y un archivo con extensión *.fig*. Estos archivos contienen toda la información básica de la interfaz, el archivo *.fig* contiene las características visuales de los elementos de cada pantalla y el archivo *.m* contiene la parte estructural de la misma, es decir, las funciones que ejecutan cada uno de los elementos que la conforman.

Es a partir de esa función principal para cada pantalla, que se hace un llamado a otras funciones o subrutinas encargadas de realizar cálculos, operaciones o construir gráficos. En el caso

de esta interfaz, realizan este llamado a los archivos MEX que conforman el modelo desarrollado, para efectuar los cálculos principales de las simulaciones.

En la interfaz se tienen siete diferentes pantallas en las que el usuario puede trabajar con las diversas opciones planteadas para las simulaciones. En ellas puede observar las propiedades y el desempeño de los códigos turbo por medio de cuatro opciones en donde se pueden variar los parámetros propios del esquema de codificación, y los parámetros propios de la simulación, que se relacionan con la validez de los resultados.

En la Fig. 2, se muestran unos resultados obtenidos utilizando la herramienta de simulación diseñada. En ellos se puede establecer la variación del desempeño de los códigos turbo, a partir de una gráfica de FER (Tasa de error de tramas) vs. E_b/N_0 , donde se han variado diferentes características de este esquema de codificación.

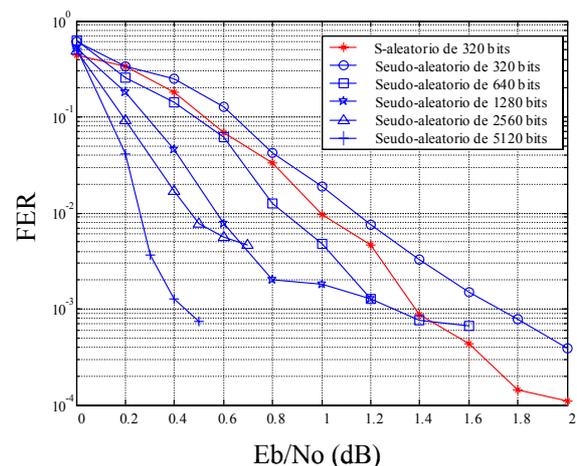


Fig. 2. Resultados obtenidos con diferentes tamaños y tipos de permutadores.

V. CONCLUSIONES

Las diferentes opciones que incluye la interfaz, la flexibilidad que se tiene en la escogencia de cada uno de los parámetros, la rigurosidad del modelo y de los algoritmos implementados, y la consistencia de los resultados obtenidos, indican que esta herramienta cumple con los requerimientos necesarios para analizar y estudiar el desempeño de los códigos turbo, presentándose como punto de partida en el desarrollo de futuros proyectos de investigación en el área de codificación de canal.

Con el desarrollo de la interfaz gráfica de esta herramienta de simulación, se presentan de forma clara y explícita cada una de las variables determinantes en el análisis de los códigos turbo, facilitando la labor de investigación de conocedores del tema, y a la vez, permitiendo su manejo a usuarios con ideas

básicas en comunicaciones que deseen conocer la estructura, el funcionamiento y las ventajas de este importante esquema de codificación.

La implementación de la función base en C++ para las simulaciones, aprovecha las principales características de eficacia de este lenguaje, mientras que el desarrollo de la interfaz en MATLAB® permite utilizar las facilidades que esta herramienta posee para la ejecución de interfaces de usuario. La unión de estas ventajas es una excelente aplicación de la alternativa que brinda MATLAB®, al permitir utilizar funciones escritas en otros lenguajes de programación.

REFERENCIAS

- [1] BERROU, Claude, GLAVIEUX, Alain, THITIMAJSHIMA, Punja. “*Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes*”, Proceedings of the IEEE International Conference on Communications, (Ginebra, Suiza), Mayo 1993, pp. 1064-1070.
- [2] MATLAB versión 6.5. “MATLAB® / Using MATLAB / External Interfaces/API”. www.mathworks.com
- [3] CÁRDENAS, Johana, TORRES, Alejandro, VÁSQUEZ, Daniel. “*Herramienta de simulación para códigos turbo*”. Tesis de grado Facultad de Ingeniería, Departamento de Electrónica, Pontificia Universidad Javeriana. Bogotá, Colombia. Mayo 2004.