Definición del problema:

Dado un mensaje de ocho bits este está representado por una función con valor 1 o 0 en cada una de 8 regiones, esta función puede ser aproximada por una suma de funciones armónicas:

“The univen distribution of energy signals in the frequency domain in the real wolrd has made signal decomposition a practical problema.” (Haddad, 1992)

por lo tanto, a la hora de enviar dicho mensaje por cierto medio, solo es necesario enviar el conjunto de ondas que corresponden con dichas funciones armónicas. Dicho lo anterior sería preferible usar el menor numero de ondas, siempre y cuando la aproximación tenga un error aceptable:

“Rate-distortion theory shows that the uneven nspectral nature of real-wolrd signals can provide the basics for source compression techniques” (Haddad, 1992)

por esto se plantea usar métodos numéricos para encontrar cual es el menor número de armónicos necesarios para tener un error aceptable.

Métodos:

Se definió:

La función como el mensaje a ser enviado en un periodo de tiempo T.

La función donde n es el numero de armónicos A, ꙍ y Ꝋ son tales que en un intervalo [0, T] esto corresponde con la serie de Fourier truncada en el enésimo armónico. Los coeficientes se calcularon:

El error E como la diferencia promedio entre las curvas, calculada integral que se aproximo por una sumatoria de Riemann por izquierda.

Luego para cada n, comenzando desde n=1 se calculo E y tan pronto como se encuentra un E menor a un valor critico dado por el usuario se detiene el proceso y se entrega muestra el n en pantalla.

Partición del programa:

Ventana.class: contiene la interfaz grafica del programa, implementada en java, se encarga de las interacciones con el usuario y hace un llamado a main.exe (archivo main.cpp compilado)

main.cpp: calcula el error de la función f y genera un archivo con sus puntos para graficarla, y hace un llamado a graph.py

Recursos.cpp: esta clase se encarga de guardar los puntos de la curva y los coeficientes de la función f.

Armonico.cpp: esta clase genera y guarda los coeficientes de la función f.

graph.py: guarda una imagen con una gráfica de la función f.

Referencias:

[Ali Akansu](https://es.wikipedia.org/w/index.php?title=Ali_Akansu&action=edit&redlink=1) and Richard Haddad, Multiresolution Signal Decomposition: Transforms, Subbands, Wavelets, Academic Press, 1992