

Evaluación y Comparación de los Algoritmos FFT.

Una vez programados y validados los algoritmos se realizaron pruebas con ellos para poderlos evaluar y comparar. A continuación se presentan dichas pruebas y sus resultados.

6.1. Evaluación del Desempeño de los Algoritmos.

Las pruebas realizadas a los algoritmos programados fueron 2:

1. Precisión de cálculo, determinada con el error promedio cuadrático y la desviación estándar del error.
2. Velocidad de cálculo en segundos y en millones de operaciones por segundo.

Todas las pruebas fueron realizadas en una computadora Dell Inspiron 1420 con procesador Intel Core 2 Duo T7250 @ 2.00 GHz. y 2 Gb. en RAM y el software Matlab 7.9.

6.1.1. Precisión de Cálculo.

Para poder calcular la precisión de los algoritmos programados se tomó como valor exacto la FFT de Matlab, para el error hacia adelante, y la señal de entrada, para el error hacia atrás. Teniendo estos valores se compararon con los resultados de cada algoritmo de dos maneras. Se obtuvo el *Error Promedio Cuadrático hacia adelante* y el *Error Promedio Cuadrático hacia atrás*. El error promedio cuadrático se calcula al comparar un valor que se considera exacto con el valor obtenido de la siguiente manera:

$$EPC = \frac{1}{N} \sum_{n=0}^{N-1} (x_r[n] - x_p[n])^2$$

En donde:

- $x_r[n]$ es el valor de la FFT de Matlab para el error hacia adelante y la señal de entrada para el error hacia atrás.
- $x_p[n]$ es el valor de la FFT de los algoritmos programados para el error hacia adelante y los algoritmos inversos programados aplicados a los resultados de los algoritmos para el error hacia atrás.
- N es el número de datos que tiene la señal de entrada.

Esta prueba se realizó tanto al módulo de la señal de respuesta como al ángulo para determinar que tan exactos son los algoritmos programados.

La señal de entrada, al igual que en la velocidad de cálculo, se trata de una señal de ruido blanco con longitud N de **2** a **4096** puntos. En la figura 6.1 se muestra un ejemplo de la señal con longitud $N = 1024$ puntos.

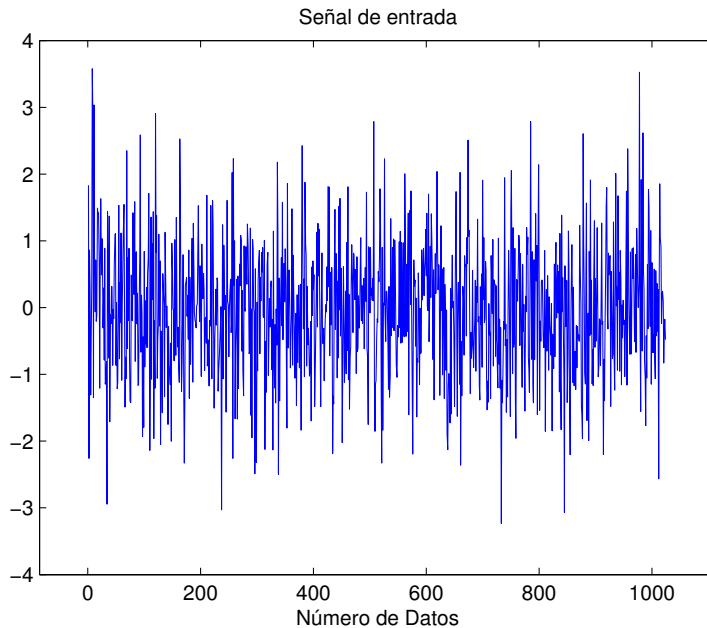


Figura 6.1: Señal de entrada de $N = 1024$ puntos.

En las gráficas 6.2 y 6.3 se presentan los resultados de los Errores Promedio Cuadráticos hacia adelante y hacia atrás del módulo de la señal respectivamente. En las tablas 6.1 y 6.2 se muestran los valores de los errores graficados.

6.1. Evaluación del Desempeño de los Algoritmos.

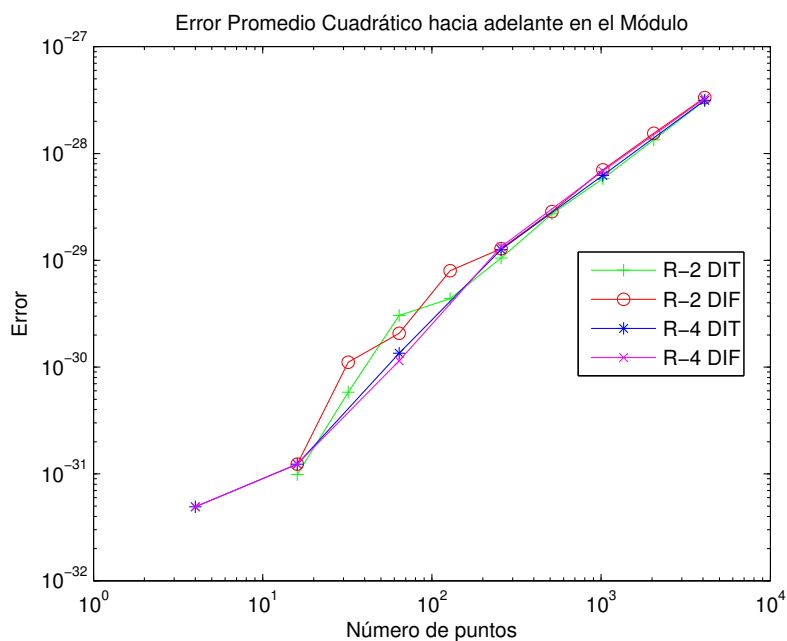


Figura 6.2: Error promedio cuadrático hacia adelante en el módulo.

N	R-2 DIT adelante	R-2 DIF adelante	R-4 DIT adelante	R-4 DIF adelante
2	0	0	-	-
4	0	0	4.9304×10^{-32}	4.9304×10^{-32}
8	0	0	-	-
16	9.8608×10^{-32}	1.2326×10^{-31}	1.2326×10^{-31}	1.2326×10^{-31}
32	5.8086×10^{-31}	1.1124×10^{-30}	-	-
64	3.0514×10^{-30}	2.0723×10^{-30}	1.347×10^{-30}	1.1442×10^{-30}
128	4.3708×10^{-30}	7.9801×10^{-30}	-	-
256	1.0498×10^{-29}	1.2858×10^{-29}	1.2622×10^{-29}	1.3381×10^{-29}
512	2.7185×10^{-29}	2.8562×10^{-29}	-	-
1024	5.8148×10^{-29}	7.0336×10^{-29}	6.2393×10^{-29}	6.8429×10^{-29}
2048	1.3398×10^{-28}	1.5467×10^{-28}	-	-
4096	3.1332×10^{-28}	3.3418×10^{-28}	3.1427×10^{-28}	3.2821×10^{-28}

Tabla 6.1: Error promedio cuadrático de los algoritmos hacia adelante en el módulo.

6.1. Evaluación del Desempeño de los Algoritmos.

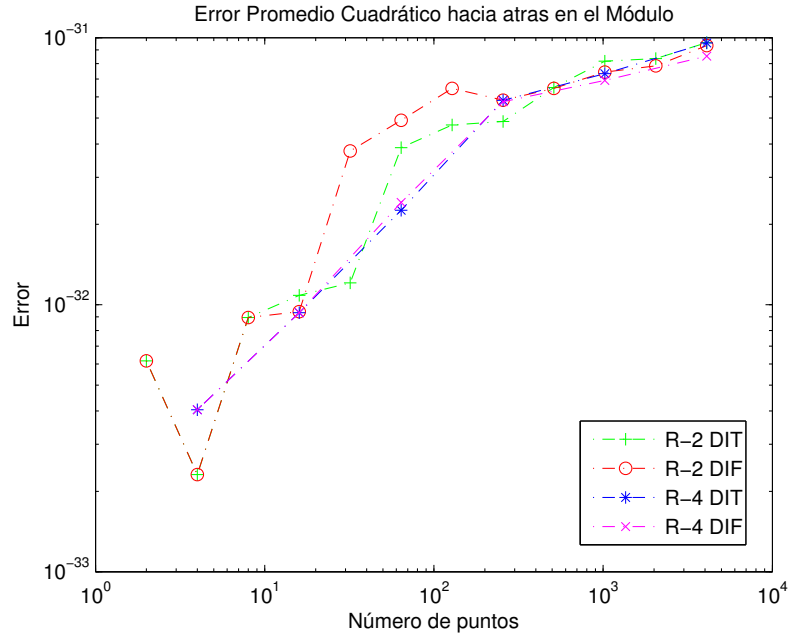


Figura 6.3: Error promedio cuadrático hacia atrás en el módulo.

N	R-2 DIT atrás	R-2 DIF atrás	R-4 DIT atrás	R-4 DIF atrás
2	6.163e-033	6.163e-033	-	-
4	2.3111e-033	2.3111e-033	4.0445e-033	4.0445e-033
8	8.9556e-033	8.9556e-033	-	-
16	1.0845e-032	9.413e-033	9.3408e-033	9.3408e-033
32	1.2075e-032	3.7665e-032	-	-
64	3.8735e-032	4.9087e-032	2.2581e-032	2.4167e-032
128	4.71e-032	6.4552e-032	-	-
256	4.8565e-032	5.8356e-032	5.8292e-032	5.757e-032
512	6.51e-032	6.4575e-032	-	-
1024	8.1676e-032	7.4304e-032	7.3299e-032	6.9194e-032
2048	8.3565e-032	7.8386e-032	-	-
4096	9.5837e-032	9.3396e-032	9.5604e-032	8.528e-032

Tabla 6.2: Error promedio cuadrático de los algoritmos hacia atrás en el módulo.

En las tablas 6.3 y 6.4 se muestran los valores de la varianza de los errores hacia adelante y hacia atrás en el módulo para cada señal de entrada con diferentes longitudes para los algoritmos programados.

6.1. Evaluación del Desempeño de los Algoritmos.

N	R-2 DIT adelante	R-2 DIF adelante	R-4 DIT adelante	R-4 DIF adelante
2	0	0	-	-
4	0	0	4.9304e-032	4.9304e-032
8	0	0	-	-
16	9.8608e-032	1.2326e-031	1.2326e-031	1.2326e-031
32	5.8086e-031	1.1124e-030	-	-
64	3.0514e-030	2.0723e-030	1.347e-030	1.1442e-030
128	4.3708e-030	7.9801e-030	-	-
256	1.0498e-029	1.2858e-029	1.2622e-029	1.3381e-029
512	2.7185e-029	2.8562e-029	-	-
1024	5.8148e-029	7.0336e-029	6.2393e-029	6.8429e-029
2048	1.3398e-028	1.5467e-028	-	-
4096	3.1332e-028	3.3418e-028	3.1427e-028	3.2821e-028

Tabla 6.3: Varianza del error para los algoritmos hacia adelante en el módulo.

N	R-2 DIT atrás	R-2 DIF atrás	R-4 DIT atrás	R-4 DIF atrás
2	6.163e-033	6.163e-033	-	-
4	2.3111e-033	2.3111e-033	4.0445e-033	4.0445e-033
8	8.9556e-033	8.9556e-033	-	-
16	1.0845e-032	9.413e-033	9.3408e-033	9.3408e-033
32	1.2075e-032	3.7665e-032	-	-
64	3.8735e-032	4.9087e-032	2.2581e-032	2.4167e-032
128	4.71e-032	6.4552e-032	-	-
256	4.8565e-032	5.8356e-032	5.8292e-032	5.757e-032
512	6.51e-032	6.4575e-032	-	-
1024	8.1676e-032	7.4304e-032	7.3299e-032	6.9194e-032
2048	8.3565e-032	7.8386e-032	-	-
4096	9.5837e-032	9.3396e-032	9.5604e-032	8.528e-032

Tabla 6.4: Varianza del error para los algoritmos en el módulo hacia atrás.

De la misma manera se realizó el cálculo del error promedio cuadrático y la varianza de éste del ángulo de fase para los algoritmos programados. El código del programa es igual al del módulo, con la diferencia del uso del comando *angle* para obtener el ángulo de fase de la señal de respuesta de los algoritmos programados.

En las figuras 6.4 y 6.5 se presentan las gráficas de los errores promedio cuadráticos hacia adelante y hacia atrás para el ángulo de fase. Y en las tablas 6.5 y 6.6 se muestran los valores respectivos de cada gráfica.

6.1. Evaluación del Desempeño de los Algoritmos.

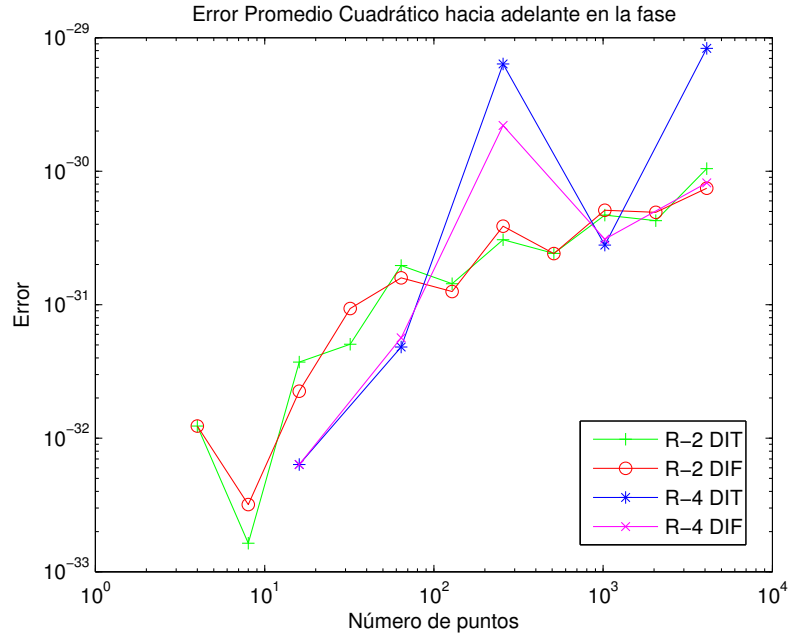


Figura 6.4: Error promedio cuadrático hacia adelante del ángulo de fase.

N	R-2 DIT adelante	R-2 DIF adelante	R-4 DIT adelante	R-4 DIF adelante
2	0	0	-	-
4	1.2326e-032	1.2326e-032	0	0
8	1.637e-033	3.1838e-033	-	-
16	3.7232e-032	2.248e-032	6.3556e-033	6.3556e-033
32	5.0556e-032	9.3696e-032	-	-
64	1.9636e-031	1.5905e-031	4.8219e-032	5.6434e-032
128	1.4361e-031	1.2561e-031	-	-
256	3.0721e-031	3.8792e-031	6.3494e-030	2.2013e-030
512	2.4198e-031	2.4181e-031	-	-
1024	4.6867e-031	5.0934e-031	2.796e-031	3.0968e-031
2048	4.2635e-031	4.9235e-031	-	-
4096	1.0462e-030	7.4557e-031	8.3298e-030	8.1742e-031

Tabla 6.5: Error cuadrático promedio de los algoritmos hacia adelante en el ángulo de fase.

6.1. Evaluación del Desempeño de los Algoritmos.

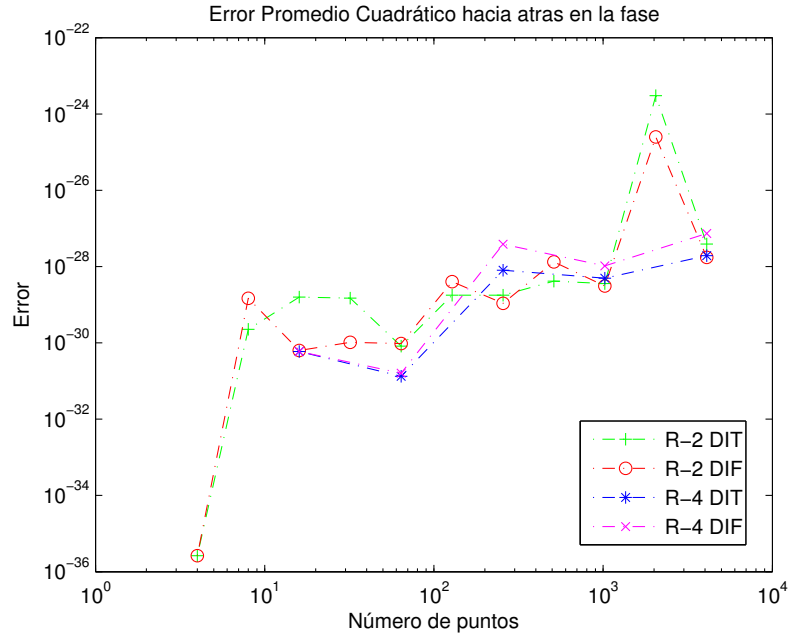


Figura 6.5: Error promedio cuadrático hacia atrás del ángulo de fase.

N	R-2 DIT atrás	R-2 DIF atrás	R-4 DIT atrás	R-4 DIF atrás
2	0	0	-	-
4	2.6267e-036	2.6267e-036	0	0
8	2.2655e-030	1.4774e-029	-	-
16	1.5921e-029	6.2911e-031	5.9211e-031	5.9211e-031
32	1.4822e-029	1.0327e-030	-	-
64	8.2379e-031	9.645e-031	1.3243e-031	1.5978e-031
128	1.7832e-029	4.0388e-029	-	-
256	1.7812e-029	1.0843e-029	8.0322e-029	3.866e-028
512	4.1467e-029	1.3197e-028	-	-
1024	3.5726e-029	3.0925e-029	4.9677e-029	1.0449e-028
2048	3.0261e-024	2.5158e-025	-	-
4096	3.9022e-028	1.7447e-028	1.934e-028	7.3825e-028

Tabla 6.6: Error promedio cuadrático de los algoritmos hacia atrás en el ángulo de fase.

En las tablas 6.7 y 6.8 se muestran los valores de la varianza de los errores hacia adelante y hacia atrás para el ángulo de fase para cada señal de entrada procesada con cada uno de los algoritmos programados.

6.1. Evaluación del Desempeño de los Algoritmos.

N	R-2 DIT adelante	R-2 DIF adelante	R-4 DIT adelante	R-4 DIF adelante
2	0	0	-	-
4	1.2326e-032	1.2326e-032	0	0
8	1.637e-033	3.1838e-033	-	-
16	3.7232e-032	2.248e-032	6.3556e-033	6.3556e-033
32	5.0556e-032	9.3696e-032	-	-
64	1.9636e-031	1.5905e-031	4.8219e-032	5.6434e-032
128	1.4361e-031	1.2561e-031	-	-
256	3.0721e-031	3.8792e-031	6.3494e-030	2.2013e-030
512	2.4198e-031	2.4181e-031	-	-
1024	4.6867e-031	5.0934e-031	2.796e-031	3.0968e-031
2048	4.2635e-031	4.9235e-031	-	-
4096	1.0462e-030	7.4557e-031	8.3298e-030	8.1742e-031

Tabla 6.7: Varianza del error para los algoritmos hacia adelante en el ángulo de fase.

N	R-2 DIT atrás	R-2 DIF atrás	R-4 DIT atrás	R-4 DIF atrás
2	0	0	-	-
4	2.6267e-036	2.6267e-036	0	0
8	2.2655e-030	1.4774e-029	-	-
16	1.5921e-029	6.2911e-031	5.9211e-031	5.9211e-031
32	1.4822e-029	1.0327e-030	-	-
64	8.2379e-031	9.645e-031	1.3243e-031	1.5978e-031
128	1.7832e-029	4.0388e-029	-	-
256	1.7812e-029	1.0843e-029	8.0322e-029	3.866e-028
512	4.1467e-029	1.3197e-028	-	-
1024	3.5726e-029	3.0925e-029	4.9677e-029	1.0449e-028
2048	3.0261e-024	2.5158e-025	-	-
4096	3.9022e-028	1.7447e-028	1.934e-028	7.3825e-028

Tabla 6.8: Varianza del error para los algoritmos hacia atrás en el ángulo de fase.

6.1.2. Velocidad de Cálculo.

La velocidad del cálculo de los algoritmos programados fue medida de 2 diferentes formas. Primero se contó el tiempo en segundos que tardan en calcular la FFT para diferentes señales de entrada con diferentes longitudes. Después se tomo ese tiempo para calcular las millones de operaciones por segundo -llamadas “mflops”- que realiza aproximadamente en el cálculo de las FFTs.

6.1. Evaluación del Desempeño de los Algoritmos.

Las señales de entrada, como en el caso de la precisión de cálculo, fueron señales de ruido blanco. La longitud de la señal se fue incrementando en cada iteración en potencias de 2 para los algoritmos Radix-2 y en potencias de 4 para los algoritmos Radix-4. La longitud de la señal de entrada va desde 2 hasta 4096 puntos.

El tiempo de cálculo se promedió para 10 iteraciones para cada longitud. Este promedio se realizó con la función *tic toc*, la cual mide el tiempo transcurrido entre *tic* y *toc*.

El tiempo transcurrido se divide entre 10 para obtener el promedio de cada FFT para las diferentes longitudes, en la gráfica 6.6 y en la table 6.9 se muestra éste tiempo en segundos.

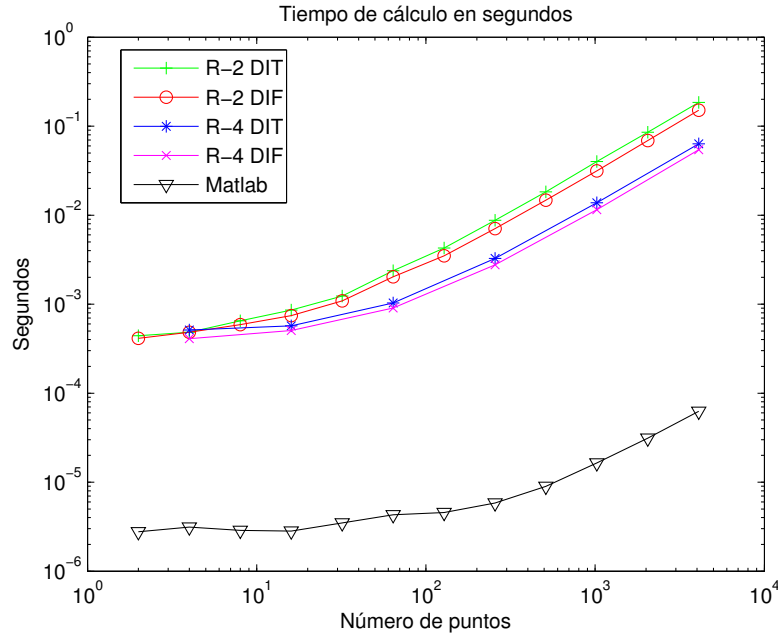


Figura 6.6: Comparación del tiempo de cálculo en segundos para los algoritmos programados y la FFT de Matlab.

N	R-2 DIT	R-2 DIF	R-4 DIT	R-4 DIF	Matlab
2	4.4274e-04	4.1333e-04	-	-	2.7716e-06
4	4.8288e-04	4.8391e-04	5.1229e-04	4.0959e-04	3.1309e-06
8	6.4995e-04	5.881e-04	-	-	2.8743e-06
16	8.6034e-04	7.4249e-04	5.7039e-04	5.0526e-04	2.823e-06
32	1.2329e-03	1.0886e-03	-	-	3.4902e-06
64	2.3736e-03	2.0325e-03	1.0333e-03	9.054e-04	4.3114e-06
128	4.282e-03	3.4899e-03	-	-	4.5681e-06
256	8.7629e-03	7.08e-03	3.2676e-03	2.7631e-03	5.8512e-06
512	1.8314e-02	1.4761e-02	-	-	8.9821e-06
1024	4.0101e-02	3.1449e-02	1.3768e-02	1.1523e-02	1.6424e-05
2048	8.5367e-02	6.8878e-02	-	-	3.1207e-05
4096	0.18451	0.15123	6.3285e-02	5.4398e-02	6.2567e-05

Tabla 6.9: Tiempo de cálculo de los algoritmos en segundos.

El cálculo de los “mflop” se realiza con la siguiente ecuación

$$mflops = \frac{5 N \log_2(N)}{\text{tiempo para una FFT en microsegundos}}$$

Donde N es el número de datos que entran en la FFT. Esta no es una cuenta real de “mflops”, es para tener una escala conveniente, basada en el hecho que el algoritmo FFT Radix-2 requiere $N \log_2 N$ operaciones de punto flotante. Esto permite comparar los diferentes algoritmos en una gráfica y proporciona una medida aproximada de la eficiencia en relación con la velocidad de reloj del sistema en el que se probó [17].

En las figuras 6.7 y 6.8 se muestran las gráficas comparativas de las velocidades de los algoritmos programados y de la FFT de Matlab. En la tabla 6.10 se muestra el número de “mflops” para cada algoritmo.

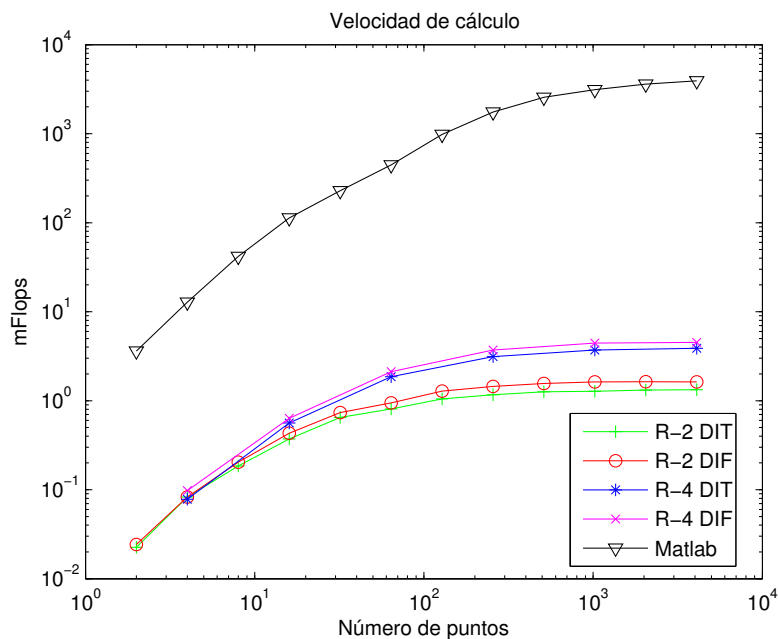


Figura 6.7: Comparación de la velocidad de cálculo en “mflop” para los algoritmos programados y la FFT de Matlab.

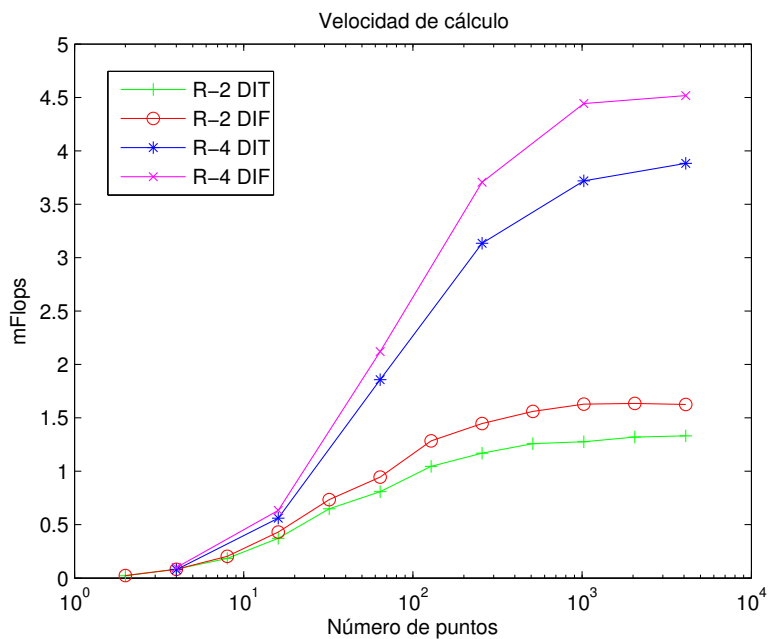


Figura 6.8: Comparación de la velocidad de cálculo en “mflops” para los algoritmos programados.

N	R-2 DIT	R-2 DIF	R-4 DIT	R-4 DIF	FFT Matlab
2	0.023043	0.024759	-	-	3.5424
4	0.084305	0.082511	0.089556	0.098523	12.5697
8	0.20281	0.21524	-	-	44.9608
16	0.39404	0.43878	0.57824	0.63997	115.4548
32	0.58763	0.67128	-	-	205.0842
64	0.88165	1.054	1.8788	2.123	456.1873
128	1.0314	1.2813	-	-	938.5359
256	1.1864	1.3606	3.1371	3.5792	1608.9187
512	1.2026	1.5781	-	-	2565.0761
1024	1.3197	1.6487	3.5049	4.2962	2977.7003
2048	1.3306	1.6518	-	-	3429.008
4096	1.3764	1.6481	3.8626	4.712	4023.6488

Tabla 6.10: Velocidad de cálculo para los algoritmos en “mflops”.

Si se comparan los algoritmos con la función de Matlab se puede observar que existe una gran diferencia entre ellos, la función `fft` de Matlab es mucho más rápida que los algoritmos programados. Esto se debe a que la función está basada en una librería llamada FFTW. La librería FFTW realiza la operación con una combinación de los algoritmos Radix-2, Radix-4, PFA y Split-Radix[18], el cual es una mezcla de los algoritmos Radix-2 y Radix-4[19].

La librería FFTW no usa un algoritmo en específico, adapta el algoritmo de la DFT a los requerimientos del usuario. El cálculo de la transformada se realiza en dos fases. Primero el planeador de la FFTW “aprende” la manera más rápida de calcular la transformada en la máquina en la que se está ejecutando. El planeador produce una estructura de datos llamada `plan` que contiene la información que “aprendió” de la máquina. A continuación se ejecuta el `plan` para realizar la transformada[17]. La FFTW puede guardar planes de transformadas previamente calculadas y de esta manera realizar los cálculos cada vez más rápido.

En el apéndice A se muestran los códigos de los programas realizados en Matlab para el cálculo de las FFTs así como los códigos de las pruebas realizadas para su comparación.

6.2. Análisis de Resultados.

De la prueba de comparación de los algoritmos programados con la función de Matlab se puede observar que prácticamente se obtienen los mismos resultados, por lo que es necesario observar los resultados obtenidos en las pruebas de precisión de cálculo.

Debido a que en la modulación OFDM para WiMAX la longitud de la FFT puede ser 128, 256, 512, 1024 y 2048, serán los valores que más se tomarán en cuenta para la toma de decisión. Se debe tomar en cuenta que el algoritmo FFT Radix-4 sólo puede ser ocupado directamente para las longitudes 256 y 1024, puesto que estos valores son potencias de 4, para los demás valores se puede realizar un ventaneo de la mitad de la longitud y así, por ejemplo, para calcular una FFT de 512 puntos se pueden calcular dos FFTs de 256 y obtener el mismo resultado. El algoritmo FFT Radix-2 puede ser utilizado sin ventaneo para las 5 longitudes pues trabaja con todas las potencias de 2.

De la figura 6.2 y la tabla 6.1 se puede determinar que el algoritmo FFT Radix-2 DIT es el que mantiene el Error Promedio Cuadrático hacia adelante con respecto al módulo más bajo de los 4 algoritmos.

De la figura 6.3 y la tabla 6.2 se puede determinar que al algoritmo FFT Radix-4 DIF cuenta con el error promedio cuadrático hacia atrás en el módulo menor para las longitudes 256 y 1024, para las longitudes restantes el algoritmo FFT Radix-2 DIF es el de mejor desempeño.

En el error promedio cuadrático hacia adelante del ángulo de fase podemos observar en la figura 6.4 y tabla 6.5 que el que demuestra mejor desempeño es el algoritmo FFT Radix-2 DIT, no siempre es el que tiene el error más bajo, pero se mantiene muy bajo sin brincos como los Radix-4.

En el error promedio cuadrático hacia atrás del ángulo de fase, en la figura 6.5 y la tabla 6.6 se puede observar que los cuatro algoritmos tiene valores muy cercanos sin mostrar que alguno tenga un mejor desempeño que los demás.

Las siguientes pruebas realizadas a los algoritmos fueron las de velocidad de cálculo. Para los valores de interés se puede observar que, para los calculables directamente por el algoritmo, el FFT Radix-4 DIF fue el más rápido. Al ser el algoritmo más rápido en tiempo de cálculo, es el que realiza más operaciones por segundo. Se puede observar que realizando 2 FFTs de la mitad de la longitud para los valores 128, 512 y 2048 sigue siendo más rápido que los algoritmos Radix-2.

6.3. Selección del Algoritmo.

Para la selección del algoritmo FFT se deben tener en cuenta los siguientes criterios: Complejidad de implementación, complejidad aritmética, precisión y velocidad de cálculo.

En cuanto a la complejidad de implementación de los algoritmos se puede decir que los Radix-4 tienen una estructura más compleja que los Radix-2. La mariposa del Radix-4

es un poco más complicada de implementar pero una vez comprendida la estructura de la mariposa y con la experiencia obtenida con la implementación del Radix-2 se simplifica.

La complejidad aritmética de los algoritmos FFT Radix-4 es menor en cuanto al número de multiplicaciones complejas realizadas para la obtención de la transformada de Fourier. Por ser menor el número de operaciones se tiene un menor tiempo de cálculo.

De las pruebas de precisión aplicadas a los algoritmos no es posible seleccionar uno de ellos para su posible implementación. Los errores mostrados en general son muy bajos, entre $1e^{-36}$ a $1e^{-28}$ para las diferentes longitudes y sin variar mucho cuando se analiza las mismas longitudes.

En cuanto a la velocidad de cálculo si existe una diferencia considerable entre los algoritmos Radix-2 y los algoritmos Radix-4, siendo el FFT Radix-4 DIF el más rápido de todos.

A partir del análisis estas pruebas se podría determinar que para WiMAX el algoritmo FFT Radix-4 DIF podría ser el más indicado para ser implementado en un procesador específico.