

2 de Maio de 2011

O Maestro

Projecto e Seminário 2010/2011

Relatório Intercalar

*Instituto Superior Engenharia de Lisboa*

Área Departamental de Engenharia de Electrónica e Telecomunicações e de Computadores

**Discentes:**  
Ana Correia, 31831

* Email : a31831@alunos.isel.pt
* Telefone: 918750435

Diogo Cardoso, 32466

* Email: a32466@alunos.isel.pt
* Telefone: 913288292

**Discentes:**  
Ana Correia, 31831

* Email : a31831@alunos.isel.pt
* Telefone: 918750435

Diogo Cardoso, 32466

* Email: a32466@alunos.isel.pt
* Telefone: 913288292

.

**Orientadores:**  
Pedro Sampaio

* Email : psampaio@cc.isel.ipl.pt

Artur Ferreira

* Email: arturj@deetc.isel.pt

..................

Índice

[1. Introdução 5](#_Toc291590941)

[1.1 Introdução ao Algoritmo Goertzel 5](#_Toc291590942)

[1.2 Goertzel vs Transformada de Fourier (FFT) 5](#_Toc291590943)

[1.3 Instrumento de estudo 6](#_Toc291590944)

[2. Algoritmo de Goertzel 7](#_Toc291590945)

[2.1 Descrição 7](#_Toc291590946)

[2.2 Características do algoritmo de Goertzel 9](#_Toc291590947)

[3. Implementação 10](#_Toc291590948)

[3.1 Introdução 10](#_Toc291590949)

[3.2 Detalhes de Implementação 10](#_Toc291590950)

[4. Testes e Resultados 13](#_Toc291590951)

[4.1 Introdução 13](#_Toc291590952)

[4.2 - 1ª Fase de Testes 13](#_Toc291590953)

[4.3 - 2ª Fase de Testes 14](#_Toc291590954)

[4.4 Tempos absolutos de Processamento 16](#_Toc291590955)

**Índice de figuras**

[Figura 1- Esquema de um Filtro de Goertzel. 8](file:///D:\FAC\LEIC\PS\working-copy\docs\relintercalar\relatóriointercalar3183132466V1.docx#_Toc291590614)

[Figura 2 - Máquina de Estados de um filtro Goertzel 10](file:///D:\FAC\LEIC\PS\working-copy\docs\relintercalar\relatóriointercalar3183132466V1.docx#_Toc291590615)

[Figura 3 - Diagrama de blocos do processamento de sinal. 11](file:///D:\FAC\LEIC\PS\working-copy\docs\relintercalar\relatóriointercalar3183132466V1.docx#_Toc291590616)

[Figura 4 - Funcionamento do Goertzel Controller 12](file:///D:\FAC\LEIC\PS\working-copy\docs\relintercalar\relatóriointercalar3183132466V1.docx#_Toc291590617)

[Figura 5 - Algumas frequências da Tabela 1. 15](#_Toc291590618)

**Índice de tabelas**

[Tabela 1 - Resultados do primeiro teste 14](#_Toc291590958)

[Tabela 2 - Resultado do calculo do tempo de processamento do algoritmo de Goertzel. 16](#_Toc291590959)

[Tabela 4 - Frequências e diferenças entre frequências de um piano 19](#_Toc291590960)

# 1. Introdução

Este documento descreve uma parte do projecto O Maestro, nomeadamente o processamento do sinal de som. O estudo necessário para realizar esta parte do projecto foi subestimado pelos elementos do grupo tendo sido necessário dominar todos os conceitos subjacentes ao algoritmo de Goertzel para ser possível implementa-lo e utiliza-lo.

## 1.1 Introdução ao Algoritmo Goertzel

O algoritmo de Goertzel (1) (2) detecta a presença de uma dada frequência através de amostragem do espectro do sinal nessa frequência. Calculado o valor do módulo do espectro de amplitude numa dada frequência e comparando-o com a energia total é possível verificar quanto é que a frequência contribuiu para a energia do sinal. Quanto menor a diferença entre a energia do sinal e a energia da frequência, maior é a contribuição da frequência para o sinal. Assim definindo um limite nesta diferença é possível avaliar se uma frequência se encontra ou não presente no sinal (3).

## 1.2 Goertzel vs Transformada de Fourier (FFT)

Quando é necessário resolver um problema que envolva detecção e processamento de frequências, normalmente, a primeira abordagem a tomar é usar a FFT. A maior diferença entre cada filtro FFT e o algoritmo de Goertzel é o facto de a FFT conseguir de uma só vez detectar várias frequências enquanto que para cada filtro de Goertzel apenas é possível detectar a presença de uma frequência.

A escolha do algoritmo de Goertzel sob a FFT deve-se ao facto de esta necessitar uma quantidade substancial de memória para funcionar, detectar todas as frequências numa dada largura de banda, ter uma elevada complexidade aritmética tornando-a mais lenta e pelo uso de valores decimais é menos portável do que o algoritmo de Goertzel.

## 1.3 Instrumento de estudo

Para testar e analisar o algoritmo de Goertzel foi necessário escolher um instrumento, nesta escolha teve-se os seguintes critérios:

* Ter uma largura de banda elevada, de forma a que fosse possível estudar o algoritmo para frequências altas e baixas.
* Ter frequências com intervalos curtos, para testar a precisão do algoritmo.

Com estes critérios escolheu-se o **piano** como instrumento de teste.

# 2. Algoritmo de Goertzel

## 2.1 Descrição

O algoritmo de Goertzel foi criado por Gerald Goertzel em 1958, este permite calcular a transformada discreta de Fourier (DFT) usando um filtro recursivo (4). Existem várias versões do algoritmo, neste documento ira-se apresentar uma versão optimizada que não tira partido de operações complexas para a detecção de frequências.

O algoritmo é composto pelos seguintes componentes:

* Um coeficiente .
* Uma constante *k* que representa a frequência que se pretende detectar.
* O valor da frequência de amostragem *Fs.*
* O valor da frequência que se pretende detectar, *Fn.*
* O numero de amostras do sinal que irão ser processadas, *N*.

O valor do coeficiente e da constante *k* são calculados pelas seguintes expressões:

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

De salientar que a constante *k* tem o valor inteiro mais próximo do resultado da equação (2).

A ilustra um filtro de Goertzel, desta pode-se deduzir a seguinte equação:

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

A equação representa o estado intermédio à medida que as n amostras "circulam" pelo filtro, esta retrata que o filtro guarda apenas os últimos dois estados intermédios para os usar posteriormente na geração de um novo.

Após o processamento de todos os elementos das N amostras o algoritmo de Goertzel retorna um valor de energia relativa através da equação .

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

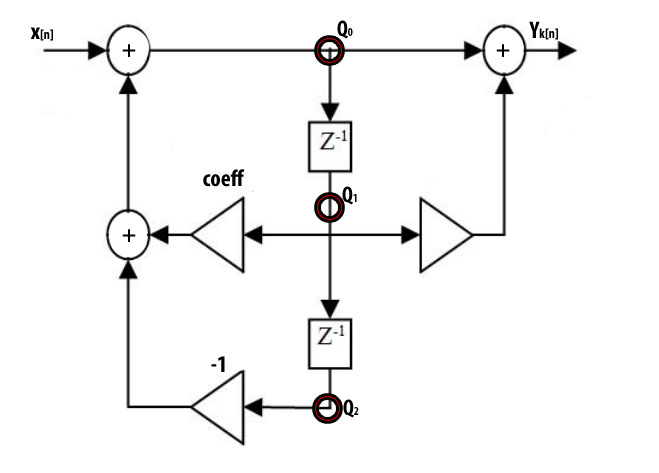


Figura - Esquema de um Filtro de Goertzel.

|  |  |  |
| --- | --- | --- |
|  |  |  |

## 2.2 Características do algoritmo de Goertzel

O algoritmo optimizado de Goertzel não usa operações complexas, além disso a sua complexidade aritmética é reduzida necessitando apenas de multiplicações e adições, sendo *N* o valor de elementos por amostra.

Em memória o algoritmo apenas necessita de ter em memória volátil as amostras a processar e os valores intermédios , e , podendo ter em memória não volátil os valores de k e coeficientes.

Outra característica do Goertzel é este ser paralelizável uma vez que cada filtro é independente de outros que possam existir, podendo assim detectar várias frequências simultaneamente.

Concluindo todos os factores referidos anteriormente tornam o algoritmo de Goertzel bastante eficiente, escalável e implementado com pouca memória, tornando-o portável a qualquer tipo de arquitectura.

# 3. Implementação

## 3.1 Introdução

Após o estudo e análise do algoritmo de Goertzel escolheu-se a linguagem C como ferramenta de implementação. A escolha desta deveu-se mais uma vez ao factor de portabilidade de código.

## 3.2 Detalhes de Implementação

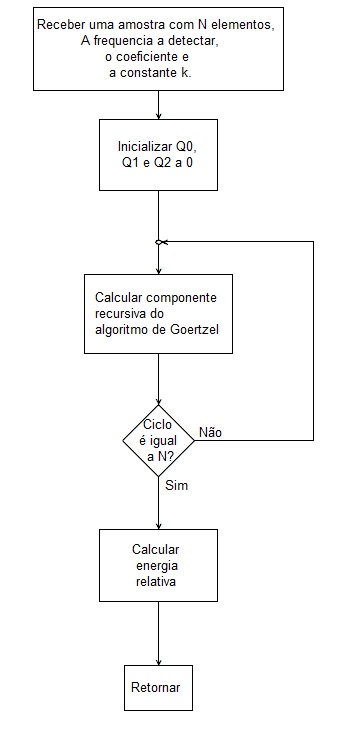
A representa a máquina de estados da implementação do algoritmo de Goertzel:

Figura - Máquina de Estados de um filtro Goertzel

Como referido anteriormente (Equação e ) este algoritmo é recursivo e necessita de apenas três variáveis locais (Q0, Q1 e Q2) para calcular o modulo do espectro de amplitude da frequência que se deseja detectar.

Durante a implementação do algoritmo teve-se de ter em conta a representação numérica das amostras, uma vez que estas deveriam ser o mais próximo possível dos cálculos teóricos. Com este factor em mente foram realizadas duas implementações, uma com valores inteiros de *32bits* e outra com valores decimais a *64bits*.

Como o algoritmo de Goertzel calcula apenas a parte positiva do espectro de amplitude (espectro bilateral) foi necessário fazer alguns ajustes de modo a que seja possível comparar o valor retornado pelo Goertzel com a energia total da amostra:

|  |  |  |
| --- | --- | --- |
|  | Goertzel  Onde *Goertzel* é o valor retornado por um filtro. | () |
|  |  | () |

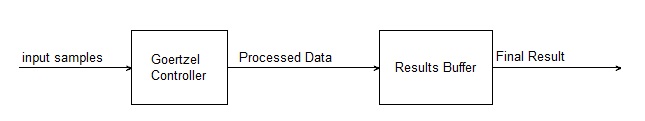
A equação (6) mostra a fórmula do cálculo do módulo do espectro de amplitude de uma dada frequência .

Figura - Diagrama de blocos do processamento de sinal.

A Figura 3 representa o pipeline de processamento de sinal utilizando o algoritmo de Goertzel, este tem apenas dois processos de manipulação de dados, o primeiro *Goertzel Controller* é responsável por analisar o sinal e referir que frequências estão presentes e o segundo, *Results Buffer* que será onde os resultados serão momentaneamente guardados de maneira a que seja possível estender futuramente as operações de processamento.

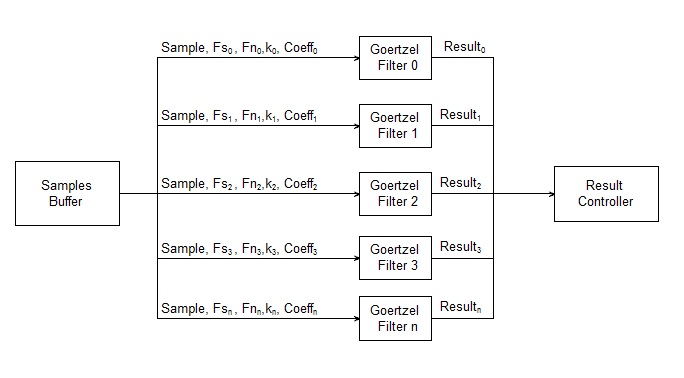
 Uma vez que o algoritmo de Goertzel é paralelizável criou-se uma infra-estrutura que tira-se partido dessa característica, de modo a controlar os seus filtros e armazenar os resultados como ilustra a Figura 4.

Figura - Funcionamento do Goertzel Controller

O *Goertzel Controller* tem como objectivo gerir filtros de Goertzel, de tal maneira que sejam processadas múltiplas frequências simultaneamente. Este também é responsável por gerir o tempo de vida de uma amostra, ou seja, quando todas as frequências tenham sido verificadas sobre uma dada amostra, esta não será mais necessária podendo portanto ser eliminada. Por fim o *Goertzel Controller* controla ainda quando é que os resultados estarão disponíveis, esta operação é crucial para que não sejam propagados resultados incompletos.

# 4. Testes e Resultados

## 4.1 Introdução

Como foi referido na introdução o instrumento de teste utilizado foi o piano, na em anexo encontra-se as frequências que pretendemos detectar bem como a diferença entre elas.

Após a implementação do algoritmo de Goertzel foi necessário testa-lo em diversos factores, nomeadamente o funcionamento básico do algoritmo sobretudo como é que programaticamente se sabe que existe uma frequência numa dada amostra. Na sequencia dos testes iniciais foram detectados outros problemas não previstos até ao momento, concretamente o comportamento do algoritmo de *Goertzel* quando se está a detectar duas frequências cuja diferença entre elas seja reduzida.   
Por fim foram realizados testes para medir o tempo que demora o algoritmo a processar as frequências desejadas.

## 4.2 - 1ª Fase de Testes

A primeira fase de testes tem como objectivo verificar o funcionamento de detecção das frequências do algoritmo de Goertzel.

Para esta fase foi realizado um modulo de criação de sinusóides de testes, que são criadas com os seguintes requisitos e o seu calculo é efectuado com a equação ):

* Frequência de Amostragem(*Fs*) = 8800 kHz
* Amplitude(*A*) = 1000
* Frequência (*fo*) é passado como parâmetro

|  |  |  |
| --- | --- | --- |
|  |  | () |

O primeiro teste a ser efectuado consistiu na criação de varias sinusóides com frequências diferentes e com que a distancia entre elas foi-se superior a 40 Hz, a mostra os resultados obtidos deste teste.

|  |  |  |
| --- | --- | --- |
| Frequências(Hz) | Percentagem do sinal(%) | Encontrado ou não |
| 55 | 21.8 | Sim |
| 110 | 22 | Sim |
| 440 | 15.1 | Sim |
| 880 | 15.6 | Sim |
| 1760 | 15.9 | Sim |
| 3520 | 16 | Sim |

Tabela - Resultados do primeiro teste

Como se pode verificar nos resultados obtidos as frequências foram encontradas mas durante o teste foram detectadas algumas falhas no algoritmo do Goertzel, principalmente no caso das frequências menores(55Hz e 110 Hz).

A principal falha foi para casos que a frequência que se pretende encontrar tenha um intervalo entre as outras frequências menor que 60 Hz, então o Goertzel considera que essas frequências vizinhas encontram-se no sinal, e como mostra a , as frequências menores são as que tem menor intervalo entre as suas vizinhas.

Também foi verificado que ao utilizar valores inteiros a 32 bits, os resultados são próximos dos resultados teóricos do algoritmo Goertzel, ou contrario, se fosse utilizado valores decimais a 64bits, mas esta escolha devesse ao *ADC* que será utilizado futuramente.

## 4.3 - 2ª Fase de Testes

Como foi referido na conclusão da 1ª Fase de Testes o algoritmo de Goertzel as frequências cujo intervalo entre elas seja curto, o Goertzel considera falsamente que existem no sinal. Esta diferença entre valores de frequências chama-se delta( este é calculado com a seguinte expressão:

|  |  |  |
| --- | --- | --- |
|  |  | () |

Por exemplo, para os valores de e temos um isso significa que caso se queira detectar uma frequência com o valor de 440Hz e que esta se encontre numa dada amostra, o algoritmo de Goertzel irá falsamente indicar que as frequências dentro do intervalo se encontram presentes no sinal, introduzindo assim um erro significativo ao processamento das amostras.

A solução ideal seria que o valor de fosse inferior a qualquer diferença entre frequências que se pretende detectar. Na encontra-se algumas frequências que se pretende detectar e a diferença entre elas.

|  |  |
| --- | --- |
| Frequência | Diferença com a anterior |
| 27,5000 | - - - |
| 29,1352 | 1,6352 |
| 30,8677 | 1,7325 |
| 32,7032 | 1,8355 |
| 34,6478 | 1,9446 |
| ... | ... |
| 3520,0000 | 197,5600 |
| 3729,3100 | 209,3100 |
| 3951,0700 | 221,7600 |
| 4186,0100 | 234,9400 |

Figura - Algumas frequências da .

Como ilustrado na , as diferenças entre as frequências são crescentes e enquanto que o anteriormente calculado era adequado para as frequências superiores a 3000Hz não o era para as frequências inferiores 740Hz sendo necessário fazer algum ajuste de modo a que o nunca seja superior à diferença de qualquer frequência.

A solução mais intuitiva seria aumentar o divisor da Equação , o N, para um valor mais próximo de Fs, por exemplo com um , o valor de seria 1 sendo inferior a todas as diferenças de frequências. O problema desta solução é que se aumentava consideravelmente o tempo de processamento do algoritmo aumentando igualmente a latência e diminuindo o tempo de resposta aos consumidores do processamento de sinal.

A segunda solução não tão evidente seria diminuir o valor de Fs, diminuindo assim também o valor de . A consequência desta solução seria que ao diminuir a frequência de amostragem estaria-se a diminuir o intervalo de frequências possíveis de serem detectadas, pelo teorema de Nyquist (2).

No final a solução adoptada foi um misto das duas anteriores, a frequência de amostragem fica constante de forma a que seja possível capturar a gama de frequências que se pretende, mas existe uma divisão desta realizada por software. Por exemplo para as primeiras frequências da o seu processamento será realizado com um e com um . Imaginando que existe um array de N posições onde são guardadas as amostras com uma frequência de amostragem de 8800*Hz*, para que os dados sejam processados com um *Fs* de 275Hz bastará que a indexação a esse array seja realizada com saltos de 34 posições uma vez que, .

Com esta solução construi-se uma aplicação utilitária que tem como funcionalidade calcular os valores de *Fs* e *N* óptimos para capturar uma dada gama de frequências.  
Por fim a 1ª Fase de Testes foi repetida agora com esta nova configuração.

## 4.4 Tempos absolutos de Processamento

Um dos aspectos mais importantes na escolha de um algoritmo de processamento de sinal é o seu tempo de processamento, para averiguar esses tempos foram realizadas algumas aplicações utilitárias que determinassem quanto tempo demoraria o Goertzel a calcular se alguma das 88 frequências estão presentes numa dada amostra, os seus resultados estão na .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Tempo Total(ns) | Tempo Relativo(ns) | Arquitectura | Processador(GHz) |
| 200 | 6022000000 | 602200 | x86 | I5 2.4 |
| 200 | 3260000000 | 326000 | x64 | I7 1.6 |
| 200 | 2792000000 | 279200 | x64 | I5 2.8 |
| 2000 | 54117000000 | 5411700 | x86 | I5 2.4 |
| 2000 | 30967000000 | 3096700 | x64 | I7 1.6 |
| 2000 | 24585000000 | 2458500 | x64 | I5 2.8 |

Tabela - Resultado do calculo do tempo de processamento do algoritmo de Goertzel.

O **tempo total** representa o tempo de execução do teste; O **tempo relativo** representa quanto tempo demora o algoritmo de Goertzel a iterar cada uma das 88 frequências de tal forma a testar se esta está presente ou não; A frequência de amostragem utilizada para estes testes foi de 8800Hz.

Os testes consistiram maioritariamente em dar uma amostra à implementação do algoritmo e esperar que ele disse-se se existia alguma frequência a detectar presente ou não.

Os testes foram executados em várias arquitecturas com diferentes processadores de tal maneira a que fosse possível visualizar como e que o algoritmo de comportava com diferentes capacidades de processamento.

## 

|  |  |  |  |
| --- | --- | --- | --- |
| Frequência | Diferença com a anterior | Frequência | Diferença com a anterior |
| 27,5000 | - - - | **349,2280** | 19,6000 |
| 29,1352 | 1,6352 | **369,9940** | 20,7660 |
| 30,8677 | 1,7325 | **391,9950** | 22,0010 |
| 32,7032 | 1,8355 | **415,3050** | 23,3100 |
| 34,6478 | 1,9446 | **440,0000** | 24,6950 |
| 36,7081 | 2,0603 | **466,1640** | 26,1640 |
| 38,8909 | 2,1828 | **493,8830** | 27,7190 |
| 41,2034 | 2,3125 | **523,2510** | 29,3680 |
| 43,6535 | 2,4501 | **554,3650** | 31,1140 |
| 46,2493 | 2,5958 | **587,3300** | 32,9650 |
| 48,9994 | 2,7501 | **622,2540** | 34,9240 |
| 51,9131 | 2,9137 | **659,2550** | 37,0010 |
| 55,0000 | 3,0869 | **698,4560** | 39,2010 |
| 58,2705 | 3,2705 | **739,9890** | 41,5330 |
| 61,7354 | 3,4649 | **783,9910** | 44,0020 |
| 65,4064 | 3,6710 | **830,6090** | 46,6180 |
| 69,2957 | 3,8893 | **880,0000** | 49,3910 |
| 73,4162 | 4,1205 | **932,3280** | 52,3280 |
| 77,7817 | 4,3655 | **987,7670** | 55,4390 |
| 82,4069 | 4,6252 | **1046,5000** | 58,7330 |
| 87,3071 | 4,9002 | **1108,7300** | 62,2300 |
| 92,4986 | 5,1915 | **1174,6600** | 65,9300 |
| 97,9989 | 5,5003 | **1244,5100** | 69,8500 |
| 103,8260 | 5,8271 | **1318,5100** | 74,0000 |
| 110,0000 | 6,1740 | **1396,9100** | 78,4000 |
| 116,5410 | 6,5410 | **1479,9800** | 83,0700 |
| 123,4710 | 6,9300 | **1567,9800** | 88,0000 |
| 130,8130 | 7,3420 | **1661,2200** | 93,2400 |
| 138,5910 | 7,7780 | **1760,0000** | 98,7800 |
| 146,8320 | 8,2410 | **1864,6600** | 104,6600 |
| 155,5630 | 8,7310 | **1975,5300** | 110,8700 |
| 164,8140 | 9,2510 | **2093,0000** | 117,4700 |
| 174,6140 | 9,8000 | **2217,4600** | 124,4600 |
| 184,9970 | 10,3830 | **2349,3200** | 131,8600 |
| 195,9980 | 11,0010 | **2489,0200** | 139,7000 |
| 207,6520 | 11,6540 | **2637,0200** | 148,0000 |
| 220,0000 | 12,3480 | **2793,8300** | 156,8100 |
| 233,0820 | 13,0820 | **2959,9600** | 166,1300 |
| 246,9420 | 13,8600 | **3135,9600** | 176,0000 |
| 261,6260 | 14,6840 | **3322,4400** | 186,4800 |
| 277,1830 | 15,5570 | **3520,0000** | 197,5600 |
| 293,6650 | 16,4820 | **3729,3100** | 209,3100 |
| 311,1270 | 17,4620 | **3951,0700** | 221,7600 |
| 329,6280 | 18,5010 | **4186,0100** | 234,9400 |

# 

Tabela - Frequências e diferenças entre frequências de um piano