# 3. Detecção de frequência

No capitulo 2 (INSERIR\_LINK) abordou-se o algoritmo de Goertzel e a FFT como possíveis soluções para a detecção de frequências, mas como descrito em (INSERIR\_LINK\_GOERTZEL\_VS\_FFT) o algoritmo de Goertzel é mais portável e menos complexo aritmeticamente do que a FFT, levando assim a que este seja o algoritmo eleito para resolução deste problema. Neste capítulo são analisadas as suas características, o seu funcionamento e algumas limitações do algoritmo.

## 3.1. - O algoritmo de Goertzel

O algoritmo de *Goertzel* foi criado por Gerald Goertzel em 1958. Este algoritmo calcula um coeficiente da transformada discreta de Fourier (*DFT* – *Discrete Fourier Transform*) através de um filtro recursivo [[4](#Rob01)]. Existem várias versões do algoritmo; neste documento trata-se uma versão optimizada que não tira partido de operações complexas para a detecção de frequências.

A ilustra o diagrama de blocos do filtro de *Goertzel*.

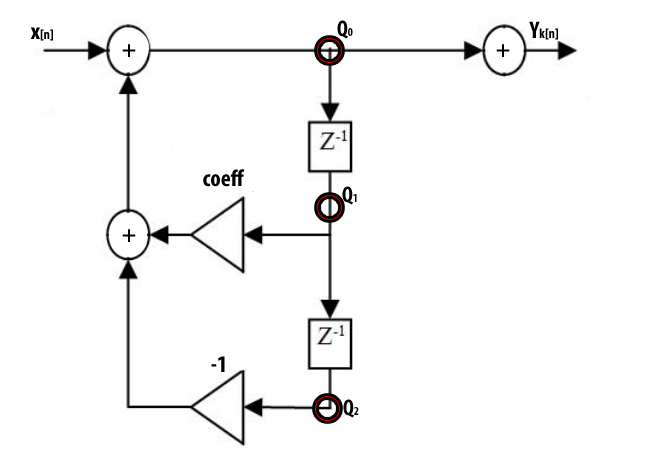


Figura - Diagrama de blocos de um filtro de Goertzel.

## 3.2. Descrição

O algoritmo de *Goertzel* [[5](#Gen1)] [[6](#MarcadorPosição1)] 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 das amostras, é possível verificar quanto é que a frequência contribui 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 [[7](#Gen111)].

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) |

A constante *k* tem o valor inteiro mais próximo resultante do arredondamento do resultado da equação (2).

A partir da pode-se deduzir as seguinte equações:

|  |  |  |
| --- | --- | --- |
|  | ;  *onde y(-1) e y(-2) = 0* | (3) |
|  |  | (4) |

A equação 3 representa a relação entre as amostras de entrada x(n) e o resultado do filtro y(n), enquanto que a equação 4 representa a evolução dos valores das unidades de atraso intermédias à medida que as N amostras "circulam" pelo filtro. 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 .

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

Na realidade o algoritmo de *Goertzel* não retorna a energia total do espectro da frequência, isto é, este só retorna o valor da energia da componente positiva do espectro. Sendo assim é necessário multiplicar por dois para obter a energia total da frequência no espectro (6).

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

Para saber se uma dada frequência está presente no sinal é necessário comparar a energia total do sinal com a energia relativa da frequência assim é necessário calcular essa energia relativa com a equação (7):

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

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

## 3.3. Características

O algoritmo optimizado de *Goertzel* não usa operações complexas, e como consequência a sua complexidade aritmética é reduzida necessitando apenas de multiplicações e adições, sendo *N* o número de amostras do sinal na entrada do filtro.

Em memória, em cada instante o algoritmo apenas necessita de ter a amostra actual 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. Através de um banco de filtros de Goertzel, é possível detectar simultaneamente a presença de várias frequências.

A resolução em frequência do algoritmo é dada pela equação (7).

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

A resolução em frequência é o intervalo entre duas frequências detectáveis, ou seja, se tivermos duas frequências *a* e *b* para detectar, a diferença entre estas deve ser maior do que o valor da resolução (8).

|  |  |  |
| --- | --- | --- |
|  | para *b > a* | (9) |

Qualquer frequência entre o intervalo ] *a , b* [ que se pretenda detectar irá ser falsamente detectada sempre que *a* ou *b* estejam presentes no sinal.

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.

## 3.4 Implementação do Algoritmo

A representa o *flowchart* da implementação do algoritmo de *Goertzel*.

Figura - Máquina de Estados de um filtro de *Goertzel*. O estado "Calcular energia relativa" refere-se à equação

A ilustra o funcionamento do algoritmo como referido anteriormente na equação e . Este algoritmo utiliza uma equação recorrente e necessita de apenas três variáveis locais (Q0, Q1 e Q2) para calcular o módulo 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 e outra com valores decimais (*floating-point*). (descritas em INSERIR\_CAP\_5\_SE)

## 3.5 Tratamento da Resolução do Goertzel

A resolução do algoritmo de *Goertzel* é dada pela equação (8) descrita anteriormente neste capitulo (INSERIR\_LIG\_PARA\_CARACT). Após a confrontação entre a gama de frequências que se pretende detectar (PAG\_DA\_TAB), a mínima frequência de amostragem(REF\_PARA\_A\_REF\_DO\_RITMO\_DE\_NYQUIST) a poder usar e os requisitos de memória conclui-se que iriam existir problemas na fase experimental do algoritmo de *Goertzel*.

Por exemplo, para os valores de e temos . Isso significa que caso se queira detectar uma frequência com o valor de 440 Hz 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 encontram-se exemplos de algumas frequências que se pretende detectar e a diferença entre as mesmas. Para baixas frequências, a necessidade de ter resolução detalhada leva a que tenha que ser utilizado um número elevado de pontos *N*.

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

Tabela - Algumas frequências da INSERIR\_REFERENCIA\_DA\_TABELA\_COMPLETA.

Como ilustrado na Tabela 1, as diferenças entre as frequências são crescentes e enquanto que a resolução anteriormente calculada era adequada para as frequências superiores a 3000 Hz não o era para as frequências inferiores a 740 Hz. Assim, foi necessário fazer ajustes de modo a que a resolução nunca seja superior à diferença entre duas frequências consecutivas.

A solução mais intuitiva seria aumentar o divisor da equação (8), 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*.

No final a solução adoptada foi um misto das duas anteriores, a frequência de amostragem fica constante para 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 275 Hz bastará que a indexação a esse *array* seja realizada com saltos de 32 posições uma vez que, .

|  |  |  |
| --- | --- | --- |
| ***Gama(Hz)*** | ***Fs (Hz)*** | ***N*** |
| **25,7 - 61,7354** | 275 | 200 |
| **65,4064 - 146,832** | 550 | 200 |
| **155,563 - 349,228** | 1100 | 200 |
| **369,994 - 830,609** | 2200 | 200 |
| **880 - 1975,53** | 8800 | 200 |
| **2093 - 4186,01** | 8800 | 100 |

Tabela - Valores de N e das frequências de amostragem para as frequências do piano.

Com esta solução construiu-se uma aplicação utilitária que tem como funcionalidade calcular os valores de *Fs* e *N* óptimos para capturar uma dada gama de frequências. Na Tabela 2 encontra-se o resultado da execução da aplicação referida anteriormente.  
   
 Com este tratamento foi possível reduzir a resolução do algoritmo de tal forma a que todas as notas sejam correctamente identificadas sem qualquer hipótese de falsas detecções pela resolução. Por exemplo a resolução para a primeira gama da ficou sendo um valor óptimo tendo em conta os valores da .

## 

## 3.6 Filtragem do sinal

O facto deste algoritmo apenas se basear no valor do coeficiente para detectar a presença de uma frequência num dado sinal, inviabiliza que existam frequências com o coeficiente igual, o problema é que ao resolver-se o problema da resolução do algoritmo(INSERIR\_LIG\_3.5), agravou-se este problema ainda mais, uma vez que a probabilidade de existirem duas ou mais frequências com o mesmo coeficiente é alta, já que as frequências estão divididas em blocos com frequências de amostragem diferentes e valor de *N* diferentes. A Tabela 3 demonstra alguns exemplos deste problema:

|  |  |
| --- | --- |
| Frequências(Hz) | Coeficiente |
| 110; 220; 440; 1760 | 0,61803 |
| 466,164; 116,54; 233,082 | 0,4743 |
| 293,665; 587,33; 2349,32 | -0,21282 |

Tabela - Exemplos de frequências com o mesmo coeficiente.

Uma vez que o problema está com os coeficientes a solução mais directa seria modular estes coeficientes até que todos os valores fossem diferentes, o problema desta solução é a complexidade de calcular coeficientes diferentes para todas as 88 frequências quando estas não partilham valores de frequências de amostragem nem de *N*. A solução terá de ser algo exterior ao algoritmo e a sua configuração, portanto optou-se por uma filtragem de sinal.

Esta filtragem irá ser realizada para cada gama de frequências () de tal maneira a que as amostras passadas ao algoritmo de *Goertzel* estejam filtradas antes deste efectuar a verificação, evitando assim as falsas detecções.

## 3.7 Filtros *FIR*



Figura - Diagrama de blocos de um filtro *FIR*

Para filtrar as amostras foram utilizados filtros do tipo FIR (Finite Impulse Response), o seu funcionamento está ilustrado na .

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

A equação representa as amostras filtradas, esta equação evidencia que o numero de amostras atrasadas presentes no filtro é o mesmo que o numero total de coeficientes .

Os coeficientes são calculados a partir da resposta impulsional de um filtro passa-banda, que por sua vez é calculado com a diferença da resposta impulsional de dois filtros passa-baixo como demonstra a equação .

|  |  |
| --- | --- |
|  | (11) |

Os valores de são dados pela equação:

|  |  |
| --- | --- |
|  | (12) |

A componente representa a frequência normalizada e é calculada com a seguinte expressão:

|  |  |
| --- | --- |
|  | (13) |

Foi ainda aplicada a janela de Hamming (INSERIR\_REF) sobre a resposta impulsional do filtro de maneira a minimizar o ganho das frequências próximas das frequências de corte, bem como foi realizada uma normalização do ganho do filtro de tal forma a que as frequências que estejam entre o filtro tenham um ganho unitário ao ser filtradas.

(INSERIR IMAGENS DE MATLAB MOSTRANDO O SINAL ANTES E DEPOIS DA JANELA E NORMALIZAÇÃO?)

(TO REMOVE? )

Por fim a aplicação descrita anteriormente em (INSERIR\_REF\_PARA\_TRATAMENTO\_DA\_RES) foi actualizada de tal forma a que todos os coeficientes dos filtros sejam automaticamente gerados.