

INSTITUTO SUPERIOR TÉCNICO

SISTEMAS DE PROCESSAMENTO DIGITAL DE SINAIS

Numerically Controlled Oscillator (NCO)

Parte I

Escrito por:

Diogo Ralo - 96921

Diogo Luís - 96922

MEE - 2022/2023

12 de março de 2023



Índice

1	Projetos de Demonstração	2
1.1	<i>Sine</i>	2
1.2	<i>Loop</i>	5
2	<i>Numerically Controlled Oscillator</i>	8
2.1	Questão 1	9
2.2	Questão 2	9
2.3	Questão 3	10
2.4	Questão 4	12
2.5	Questão 5	13
2.6	Questão 6	13
2.7	Questão 7	15
2.8	Questão 8	16
2.9	Questão 9	20

Capítulo 1

Projetos de Demonstração

O objetivo principal deste trabalho é familiarizar com sistemas de desenvolvimento de *software* e de processamento digital de sinais. Para tal será usado o kit de desenvolvimento DSK TMS320C5515 da *Texas Instruments*.

1.1 *Sine*

Analizando o ficheiro `main.c` do projeto **Sine**, é possível observar a inicialização e configuração do codec de áudio AIC3204 na qual, entre outros parâmetros, é definida a frequência de amostragem, $f_s = 16$ kHz. Para gerar uma sinusoide, é inicializada uma *Look-Up Table* (LUT) com 16 valores como se demonstra abaixo.

```
//Sinewave Look-Up Table
Int16 SineWave[] = {0, 383, 707, 924, 1000, 924, 707, 383,
                    0, -383, -707, -924, -1000, -924, -707, -383};
```

Usando esta LUT como um *buffer* circular, é possível verificar à saída um sinal sinusoidal como o que está representado na figura 1.2. Antes de fornecer os valores da `SineWave[]` aos DACs, estes são ainda multiplicados pela variável `gain`, o que se traduz numa amplificação digital do sinal. No caso das figuras 1.1 e 1.2, o ganho tem o valor 32.

Como o sinal é formado por 16 níveis transmitidos a cada $T_s = \frac{1}{f_s}$, a sinusoide gerada à saída do DSK tem frequência $f_{out} = \frac{f_s}{16} = 1$ kHz.

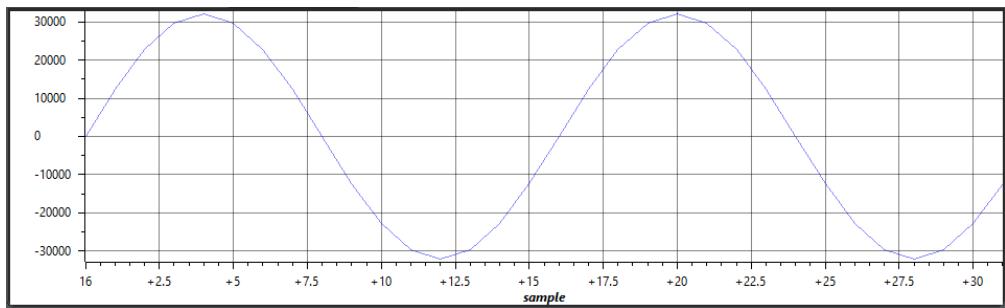


Figura 1.1: Gráfico do CCS para `gain = 32`

Relativamente à figura 1.2, os oscilogramas adquiridos no canal CH1 (a amarelo) e no canal CH2 (a azul), referem-se respetivamente às componentes direita e esquerda do canal de saída.

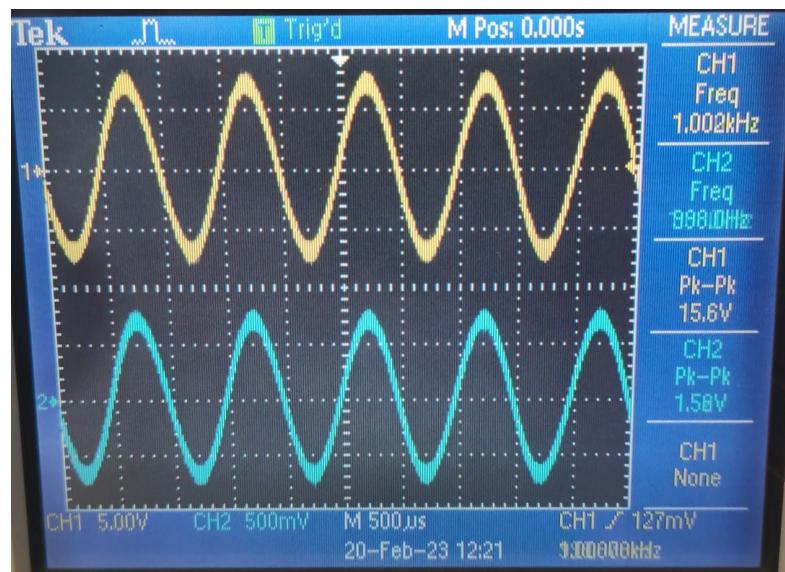


Figura 1.2: Oscilograma do sinal adquirido para `gain = 32`

De seguida, alterou-se o valor da variável `gain` de 32 para 33 e voltou-se a executar o programa, tendo-se obtido os gráficos representados pelas figuras 1.3 e 1.4.

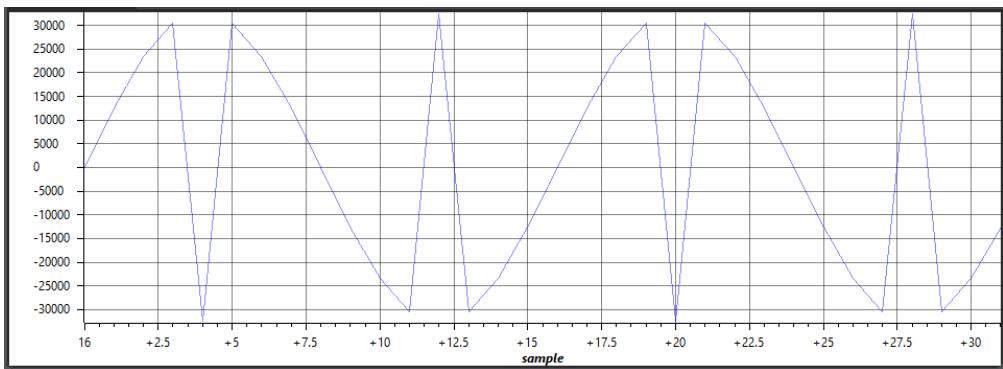


Figura 1.3: Gráfico do CCS para `gain=33`

A partir do gráfico acima, observa-se que o sinal gerado pelo DSK difere de uma sinusóide, sendo que, após o produto pelo ganho, os pontos máximos e mínimos da LUT `SineWave[]` fazem corresponder a valores diferentes do esperado. Isto deve-se ao facto de que certos valores resultantes desse produto (guardados no array `SineBuffer[]`) não podem ser representados em complemento para dois com 16 bits. Deste modo, apenas é possível representar números inteiros entre -32768 e 32767 e o não cumprimento desta condição resultará num erro de *overflow*.

$$\begin{aligned}
 \boxed{\text{SineWave[4] * gain}} &= 1000 \times 33 \\
 &\Downarrow \\
 33000 \longrightarrow \dots &0000 \textcolor{red}{1000} \textbf{0000} \textcolor{black}{1110} \textbf{1000} \longrightarrow -32536 \\
 &\quad (0x80E8) \\
 &\Downarrow \\
 \boxed{\text{SineBuffer[4] = -32536;}}
 \end{aligned}$$

$$\begin{aligned}
 \boxed{\text{SineWave[12] * gain}} &= -1000 \times 33 \\
 &\Downarrow \\
 -33000 \longrightarrow \dots &1111 \textcolor{red}{0111} \textbf{1111} \textcolor{black}{0001} \textbf{1000} \longrightarrow +32536 \\
 &\quad (0x7F18) \\
 &\Downarrow \\
 \boxed{\text{SineBuffer[12] = 32536;}}
 \end{aligned}$$

Para reproduzir um sinal idêntico ao do gráfico da figura 1.3, são necessárias componentes a frequências mais altas, que excedem a frequência de corte do filtro passa-baixo à saída do codec. Por este motivo, o sinal obtido no osciloscópio aparenta ser um pouco diferente do esperado de acordo com o gráfico gerado pelo *Code Composer Studio*.

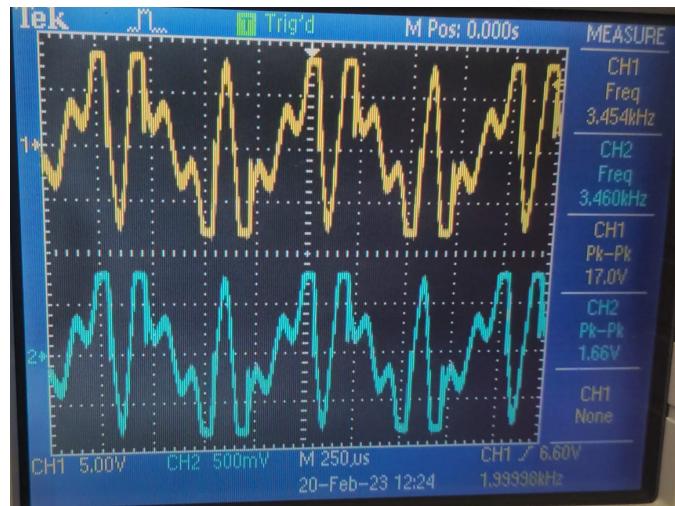


Figura 1.4: Oscilograma do sinal adquirido para gain = 33

1.2 Loop

Neste trabalho, o programa consiste em adquirir sinais dos canais de entrada do DSK e retransmiti-los nos canais de saída. Para tal, as amostras adquiridas pelos conversores analógicos-digitais são transmitidas aos conversores digitais-analógicos.

```
DataOutLeft = DataInLeft;           // loop left channel samples
DataOutRight = DataInRight;         // loop right channel samples
```

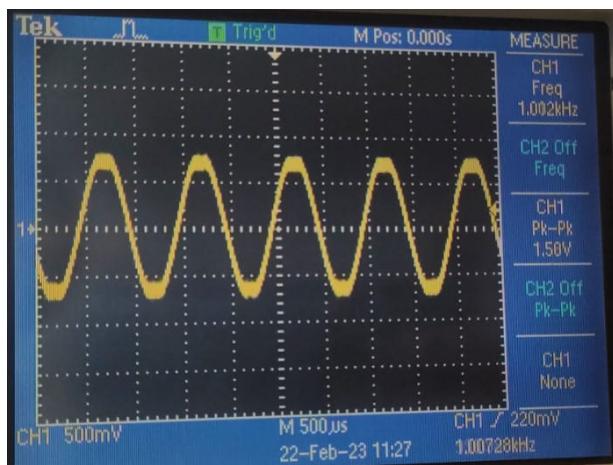


Figura 1.5: Entrada Sinusoidal

A figura 1.5 apresenta o gráfico do sinal de entrada à saída do codec. Como é possível verificar a frequência e a amplitude correspondem às do sinal de entrada (configurado no gerador de funções para produzir uma sinusoide com 1.5 V pico a pico e frequência de 1 kHz). No entanto, é também possível observar que nos máximos e mínimos do sinal observado no osciloscópio, verifica-se um pequeno ”achatar” do gráfico. Tal pode ser explicado pelo facto de os ADCs saturarem

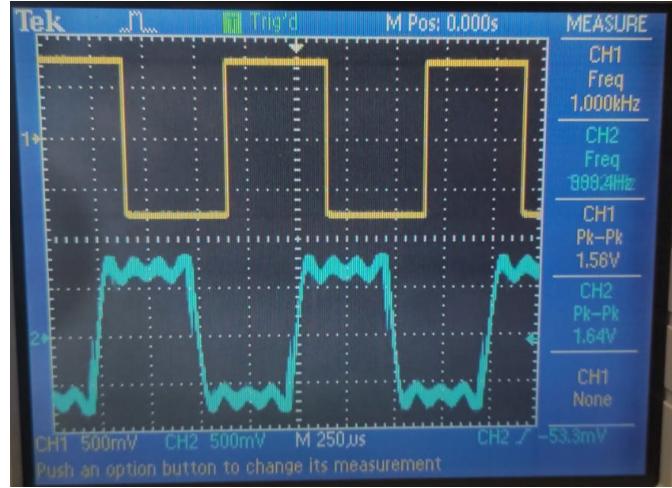


Figura 1.6: $f_0 = 1$ kHz

A figura 1.6 contém os gráficos dos sinais de entrada (representado a amarelo que corresponde a uma onda quadrada com $f = 1$ kHz) e o sinal de saída do DSK (representado a azul com a mesma frequência do sinal de entrada). É possível também observar que o sinal de saída não corresponde exatamente a uma onda quadrada e isso deve-se ao facto de o filtro presente na saída impedir (ou ”cortar”) a passagem de parte das harmónicas necessárias para a correta representação de uma onda quadrada (que teoricamente são infinitas, neste caso às frequências 3, 5, 7, 9, ... kHz). Além disso, destaca-se também o atraso do sinal de saída relativo ao de entrada, atraso esse que pode ser explicado pelo tempo de conversão dos conversores (tanto analógico-digitais como digitais-analógicos).

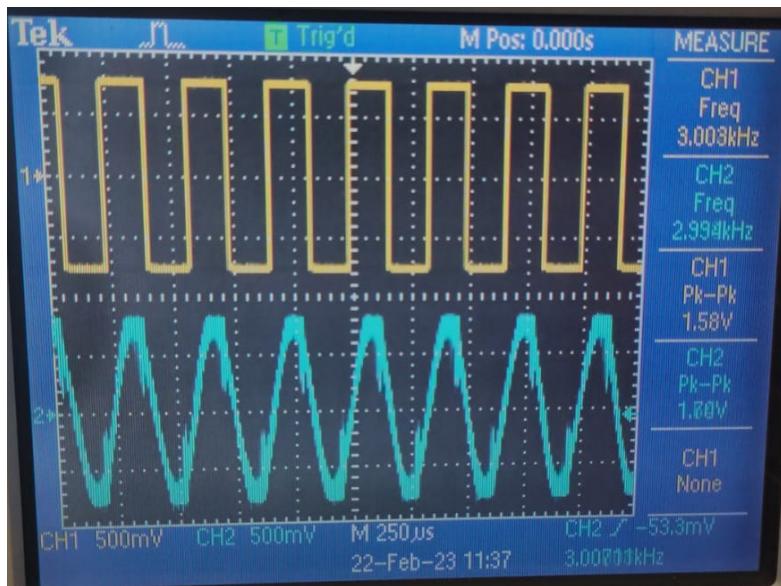


Figura 1.7: $f_0 = 3$ kHz

Na figura 1.7, à semelhança da figura 1.6 temos também o sinal obtido à saída do gerador de sinais (sendo que neste caso alterou-se a frequência que passou a ser 3 kHz) e o sinal de saída do DSK. Por comparação das duas figuras, é possível verificar que o sinal de saída teve gráficos drasticamente diferentes para os dois valores de frequências do sinal de entrada (não se alterando apenas a frequência do primeiro). Este comportamento deve-se ao facto de o espetro do sinal de entrada apresentar harmónicas nos múltiplos ímpares da frequência fundamental (neste caso 3 kHz), ou seja, o espectro tem componentes não nulas nas frequências $m \times f_0$, $m \in \{1, 3, 5, 7, 9, \dots\}$. Concretizando, a segunda harmónica estará nos 9 kHz, a terceira nos 15 kHz e assim sucessivamente. Desta forma, e sabendo que o filtro passa-baixo que está à saída do DSK tem uma frequência de corte próxima dos 8 kHz, é de esperar que depois de passar no filtro, o espetro do sinal de entrada tenha apenas uma componente nos 3 kHz (que corresponde à frequência fundamental), já que todas as outras componentes serão eliminadas pelo filtro (a segunda harmónica está centrada nos 9 kHz). Por esta razão, o sinal de saída apresenta um gráfico manifestamente diferente do sinal de entrada, uma vez que o seu espetro apresenta apenas componentes numa frequência, componente essa que é insuficiente para representar uma onda quadrada. Para além disso, observa-se também uma desfasagem entre os sinais, cujas causas foram explicadas anteriormente.

Capítulo 2

Numerically Controlled Oscillator

Este oscilador corresponde à parte fulcral da primeira entrega deste projeto e, segundo o enunciado do mesmo, tem de cumprir as especificações presentes na tabela 2.1.

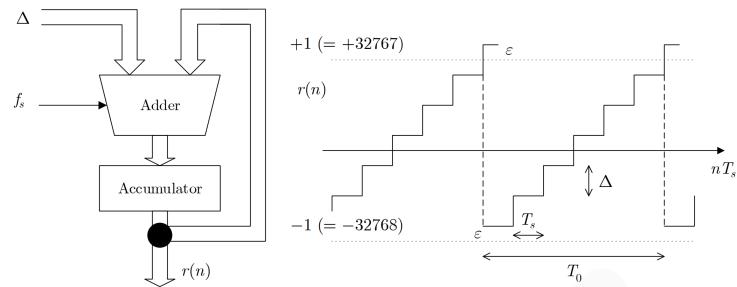


Figura 2.1: Princípio de funcionamento do oscilador de relaxação

Na figura 2.1 apresenta-se o princípio e forma de onda de um oscilador de relaxação. Como é possível verificar a variável Δ controla a frequência do oscilador, já que ao aumentar o valor da

Tabela 2.1: Especificações do oscilador de relaxação

Parâmetro	Símbolo	Valor	Reparo
Frequência de amostragem	$f_s = \frac{1}{T_s}$	16 kHz	Pré-programada
Frequência mínima	f_{min}	2 kHz	Frequência à qual a amplitude do sinal de entrada é mínima
Frequência máxima	f_{max}	6 kHz	Frequência à qual a amplitude do sinal de entrada é máxima
Canal direito de saída			Saída do oscilador, $r(n)$
Canal esquerdo de saída			Saída sinusoidal do oscilador, $y(n)$

mesma o valor de T_0 irá diminuir.

2.1 Questão 1

Para criar um oscilador de relaxação, basta incrementar o valor $r(n)$ pelo valor Δ a cada intervalo de amostragem. Como o valor de $r(n)$ é guardado numa variável do tipo *short*, cuja dimensão é 16 bits, haverá eventualmente *overflow*, limitando a variável `ramp` a valores dentro do intervalo $[-32768; 32767]$. Devido ao comportamento circular da representação em complemento para dois, não é necessária nenhuma operação adicional. Ou seja, o número imediatamente a seguir a 32767 (0x7FFF) é -32768 (0x8000) e, devido ao *overflow*, o número imediatamente a seguir a -1 (0xFFFF) é 0 (0x0000).

```
void main(void) {
    //...
    short ramp, delta = DELTA0;
    while(1) {
        //...
        ramp += delta;
        DataOutRight = ramp;
    }
}
```

2.2 Questão 2

Nesta questão é pedido para calcular o valor de Δ e uma expressão que relate a frequência do sinal de saída com Δ e f_s . Foram então realizadas as diversas etapas que se apresentam de seguida.

É necessário salientar que:

- T_0 corresponde ao período do sinal de saída do oscilador
- T_s representa o período de amostragem
- N corresponde ao número de níveis, no caso, $N = 2^{16}$
- Δ corresponde ao valor representado na figura 2.1

$$\frac{T_0}{T_s} = \frac{N}{\Delta} \implies \frac{f_s}{f_0} = \frac{N}{\Delta} \implies f_0 = \frac{f_s}{N} \Delta$$

Permitindo chegar à expressão pedida:

$$\Delta = \frac{f_0}{f_s} N \quad (2.1)$$

Para além disso, é também pedido que se determine o Δ necessário para obter f_0 , ou seja Δ_0 . Substituindo os respetivos valores em 2.1 obtemos:

$$\Delta_0 = \frac{4 \cdot 10^3}{16 \cdot 10^3} \cdot 2^{16} = 16384$$

Utilizando o mesmo raciocínio para as frequências máximas e mínimas:

$$\Delta_{min} = \frac{2 \cdot 10^3}{16 \cdot 10^3} \cdot 2^{16} = 8192$$

$$\Delta_{max} = \frac{6 \cdot 10^3}{16 \cdot 10^3} \cdot 2^{16} = 24576$$

Com estes resultados, são definidas as seguintes macros para os valores de Δ calculados:

```
#define DELTA0 0x4000      //=16384 --> f0 = 4kHz
#define DELTA_MIN 0x2000    //=8192 --> fmin = 2kHz
#define DELTA_MAX 0x6000    //=24576 --> fmax = 6kHz
```

2.3 Questão 3

O sinal adquirido no osciloscópio a partir do canal de saída direito do DSK encontra-se representado nas figuras abaixo para as frequências de oscilação: $f_0 = 4$ kHz (fig. 2.2), $f_{min} = 2$ kHz (fig. 2.3a) e $f_{max} = 6$ kHz (fig. 2.3b). É de notar que o oscilograma apresenta um sinal dente-de-serra decrescente, ao contrário do esperado, devido a um circuito inversor presente no kit de desenvolvimento. Para além disso, pode-se observar que quanto maior a frequência do oscilador, mais o seu comportamento difere do esperado. Isto deve-se, analogamente ao caso anterior, ao filtro passa-baixo na saída do codec, que corta as harmónicas acima de 8 kHz. Neste caso, tratando-se de um sinal do tipo dente-de-serra, que não tem simetria de meia-onda, o seu espetro original apresentaria harmónicas em todos os múltiplos da frequência fundamental.

Por exemplo, no caso correspondente à frequência f_0 , o sinal observado tem componentes na frequência fundamental, 4 kHz, e na primeira harmónica, 8kHz (atenuada pois coincide com a frequência de corte do filtro). Por outro lado, o oscilador a 2 kHz tem 3 harmónicas (4, 6 e 8 kHz) assemelhando-se mais ao gráfico da figura 2.1.

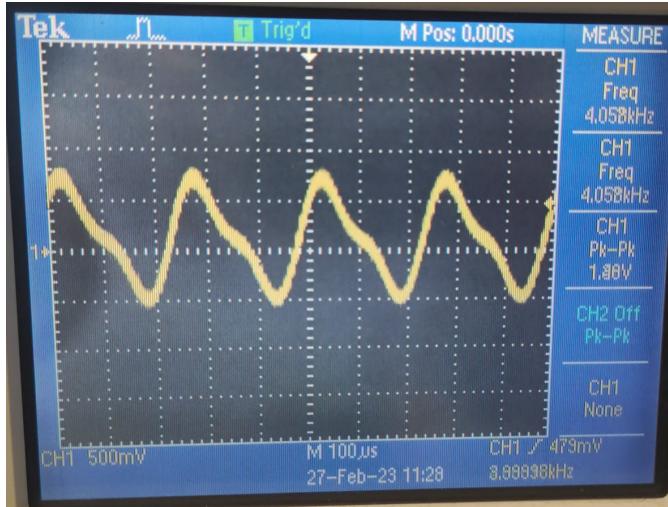
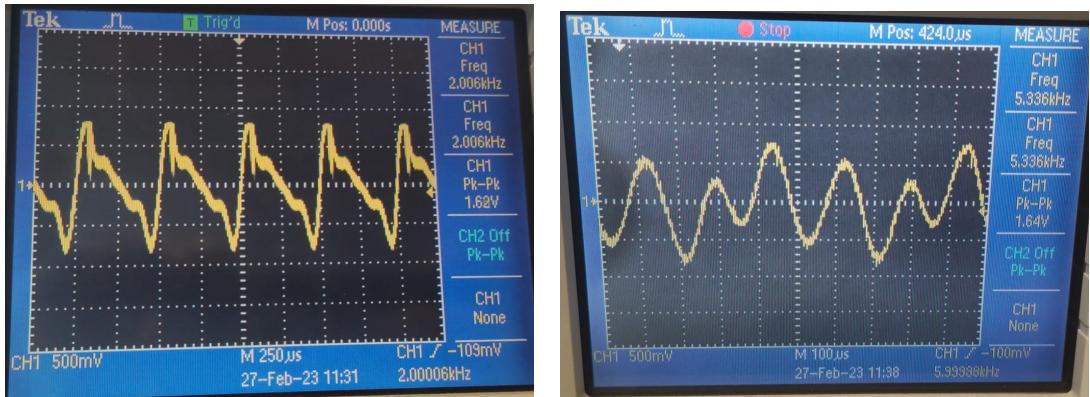


Figura 2.2: Oscilador de relaxação com $f_0 = 4$ kHz



(a) Oscilador de relaxação com $f_{min} = 2$ kHz

(b) Oscilador de relaxação com $f_{max} = 6$ kHz

Figura 2.3

Já no caso da figura 2.3b, com o oscilador à frequência máxima, acontece um fenômeno diferente. Ao contrário dos oscilogramas anteriores, este difere bastante do comportamento esperado de um oscilador de relaxação. Isto acontece porque o valor de Δ_{max} não é um divisor de 2^{16} , causando que o período de *overflow* de `ramp`, T_o , não corresponda com a periodicidade do sinal. Para determinar o período fundamental de $r(n)$, calcula-se o mínimo múltiplo comum $mmc(\Delta_{max}, 2^{16}) = 2^{16} \times 3$, o que indica que o seu valor é o triplo de T_o e, consequentemente, a frequência fundamental é $f_{max}/3 = 2$ kHz. Isto resulta que o sinal tenha a DFT que se apresenta de seguida (na figura 2.4), na qual a componente dominante não corresponde à frequência fundamental, causando o comportamento diferente do previsto.

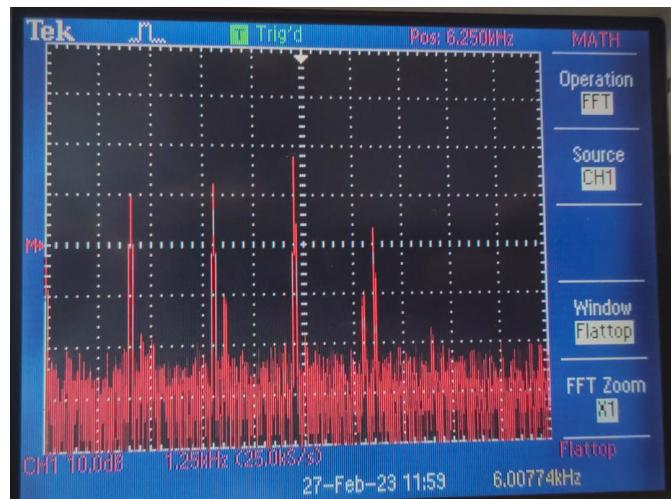


Figura 2.4: DFT do oscilador com $f_0 = 6$ kHz

Na figura 2.4, também se pode evidenciar o efeito do filtro passa-baixo, já que a harmónica dos 8 kHz está atenuada e as harmónicas em frequências mais altas são rejeitadas.

É importante destacar que a distorção observada nos sinais das figuras 2.2 e 2.3 apenas se verifica na emissão do respetivo sinal nos canais de saída do DSK. Do ponto de vista do processador digital de sinal (DSP), o oscilador tem um comportamento igual ao descrito pelo diagrama 2.1, consistindo num sinal dente-de-serra crescente, discretizado no tempo e na amplitude.

2.4 Questão 4

Nesta questão é pedido para definir uma LUT (*Look-up table*) com 32 posições, em que cada uma contém um valor de metade do ciclo de uma sinusoide representado no formato Q_{15} . A expressão para a obtenção destes valores é

$$y(m) = \text{round}\left(32767 \cdot \sin\left(\frac{\pi}{32}m\right)\right), \quad \text{com } m = 0, 1, \dots, 31$$

A expressão acima permitiu calcular os valores da LUT, tendo-se obtido

```
Int16 lut[32] =
{0,3212,6393,9512,12539,15446,18204,20787,23170,25329,27245,28898,30273,31356,
32137,32609,32767,32609,32137,31356,30273,28898,27245,25329,23170,20787,18204,
15446,12539,9512,6393,3212};
```

2.5 Questão 5

Nesta questão é pedido para usar a variável de estado r para indexar os diferentes valores da LUT.

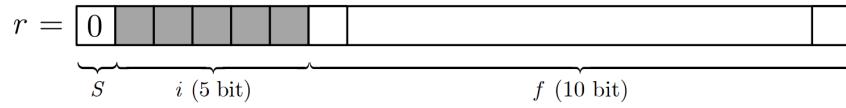


Figura 2.5: Decomposição da variável r

Por análise da figura 2.5 é possível concluir que esta pode ser decomposta $r = (-1)^S \cdot i.f$, em que S representa o bit de sinal, i os 5 bits que contém informação relativamente à parte inteira do valor guardado em r e f corresponde aos 10 bits da parte fracionária.

Assim, a saída sinusoidal do DSK resultará da expressão $y(n) = LUT(i)$. Há ainda que ter em conta que os valores da LUT correspondem apenas a meio ciclo da sinusoide, sendo que deve ser feita uma gestão do sinal de y conforme o bit S . Ou seja, valores negativos de $r(n)$ (`ramp`) fazem corresponder a valores negativos de $y(n)$ (`y`), e vice-versa. A implementação foi feita da seguinte forma:

```
//...
i = (ramp>>10) & 0x001F;
y = lut[i];
if (ramp < 0){
    y = -y;
}

DataOutLeft = y;
//...
```

2.6 Questão 6

Na figura 2.6 apresentam-se o sinal do oscilador de relaxação, $r(n)$ (canal CH1), e a sinusoide, $y(n)$ (canal CH2), gerada a partir da LUT.

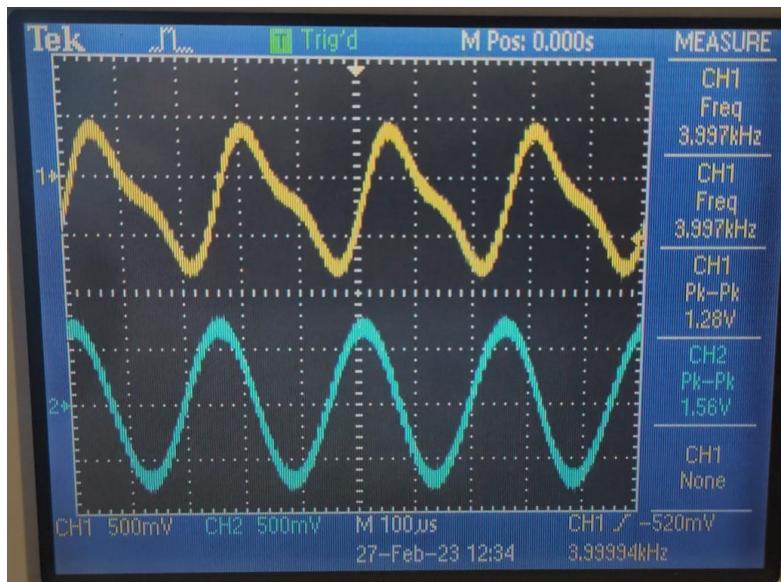


Figura 2.6: Sinais de saída do DSK para um ganho unitário

Foi ainda criada a variável `gain` do tipo `Int16` para o controlo da amplitude de $y(n)$. O produto do ganho pelo valor da LUT é feito em representação de ponto fixo, na notação Q_{15} , visto que ambos os operandos representam valores entre -1 e 1.

```
gain = 0x4000; // 16384 ---> 1/2 in Q15
y = (((long)lut[i] * gain)<<1)>>16;
```

A multiplicação entre dois números de 16 bits resulta num número de 32 bits, logo, é necessário fazer o *cast* de um dos operandos para essa dimensão (`((long)lut[i])`). O resultado da operação de multiplicação é um número em Q_{30} com um bit de sinal redundante e, fazendo um *shift* para a esquerda (`<<1`), é possível eliminar esse bit redundante e representar o valor em Q_{31} . O *shift* de 16 para a direita (`>>16`) serve para truncar o resultado para 16 bits.

No exemplo acima, usa-se um ganho de $\frac{1}{2}$, portanto, como se pretende representar o seu valor em notação Q_{15} , a variável `gain` é inicializada com $\frac{1}{2} \cdot 2^{15} = 16384 = 0x4000$. O resultado está apresentado na figura 2.7.

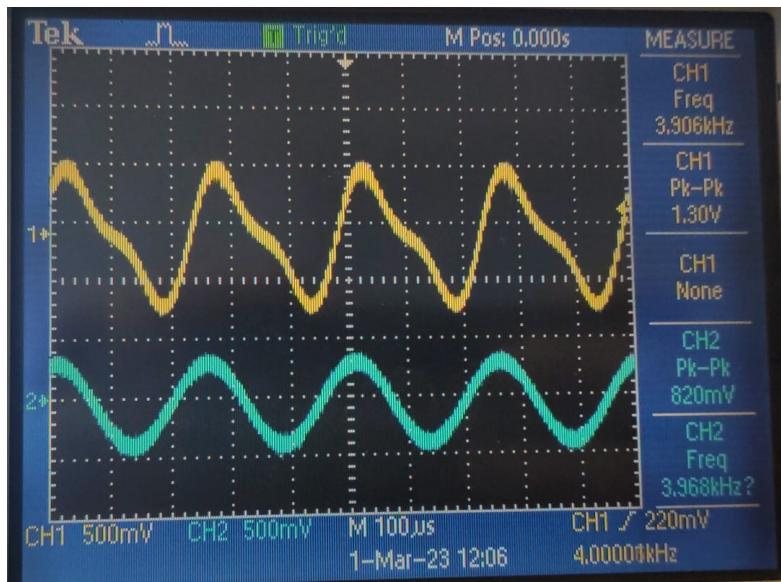


Figura 2.7: Sinais de saída do DSK para um ganho de 0.5

2.7 Questão 7

Daqui em diante, para calcular o valor de Δ , será utilizada a equação dada no guia deste projeto e que se apresenta de seguida:

$$\Delta = \Delta_0 + k \cdot \varepsilon(n) \quad (2.2)$$

É necessário referir que na equação 2.2, k é uma constante que é determinada para o oscilador ter a gama de frequências pedida. O sinal $\varepsilon(n)$ controlará a frequência do sinal de saída do oscilador.

Sabendo que $\varepsilon(n) \in [-32768; 32767]$, foram realizadas as seguintes etapas

$$\Delta_{max} = \Delta_0 + k \cdot 32767 \iff k = \frac{\Delta_{max} - \Delta_0}{32767} \iff k = 0,25 = \frac{1}{4}$$

O produto pode ser feito utilizando representação em ponto fixo, no entanto, como k é uma potência de base 2, o cálculo de Δ pode ser feito da seguinte forma, que se mostra mais simples e rápida.

$$\Delta = \Delta_0 + (\varepsilon(n) >> 2) \quad (2.3)$$

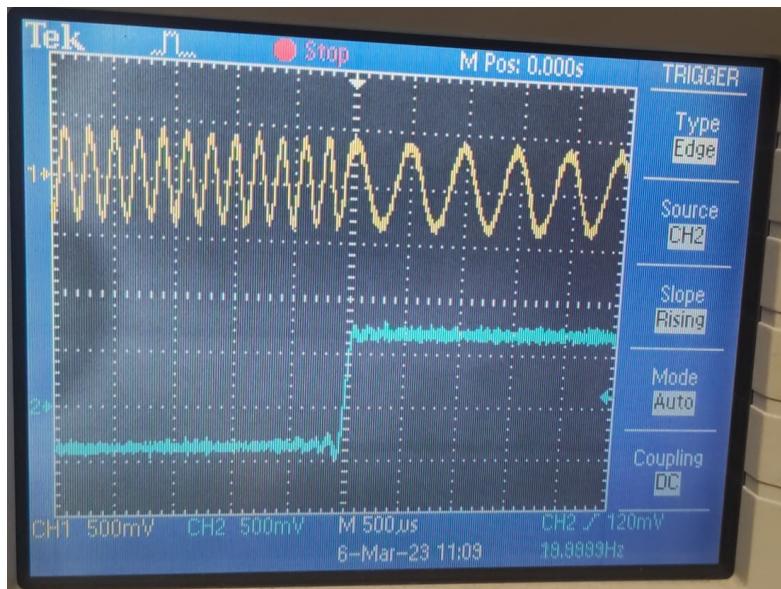


Figura 2.8: Sinal do gerador amostrado e sinal de saída do DSK modulado em frequência

Na figura 2.8 é possível observar tanto o sinal do gerador de funções depois de ser amostrado pelo DSK (no CH2, a azul), bem como o sinal modulado em frequência pelo mesmo (no CH1, a amarelo). É necessário referir dois tópicos, um sobre cada sinal. Começando pelo sinal amostrado, é visível que não corresponde a uma onda quadrada perfeita (neste caso o foco será as pequenas oscilações próximas da transição e não na transição que já seria de esperar que não tivesse declive infinito). Estas pequenas oscilações devem-se ao filtro passa-baixo presente à saída do codec, que filtra as componentes superiores a 8 kHz. Para além disso, seria de esperar que ao menor valor do sinal gerado externamente ao DSK correspondessem frequências mais baixas e ao valor máximo do sinal originado no gerador correspondessem frequências mais elevadas. Por observação, verifica-se que não é isso que acontece e esse facto pode ser explicado pelo circuito inversor presente dentro do DSK (como foi referido anteriormente neste relatório).

2.8 Questão 8

Com vista a melhorar a qualidade do oscilador sinusoidal, aplica-se interpolação linear usando a parte fracionária da variável `ramp`.

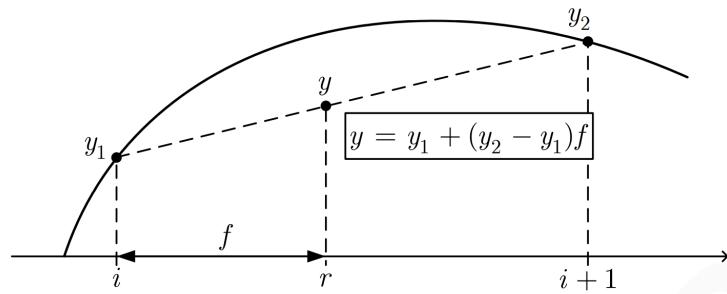


Figura 2.9: Esquema explicativo da interpolação linear

A interpolação é implementada conforme se demonstra abaixo:

```

f = (ramp<<5) & 0x7FFF;
y1 = lut[i];
y2 = lut[i+1];

y = (((long)(y2-y1) * f)<<1)>>16;
y = y1 + y;

if (ramp < 0){
    y1=-y1;
}
DataOutRight = y1;

```

De forma a melhor visualizar as diferenças entre o sinal de saída do oscilador com e sem interpolação para duas frequências diferentes ($f_0 = 4\text{kHz}$ e $f_0 = 3.9991\text{kHz}$) é pedido para utilizar a funcionalidade de FFT (*Fast Fourier Transform*) do osciloscópio.

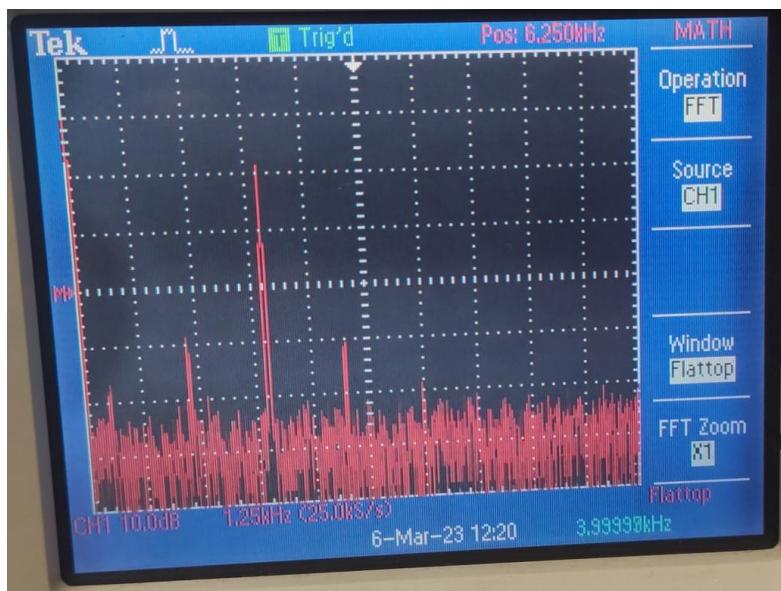


Figura 2.10: FFT do sinal de saída interpolado a $f_0 = 4\text{kHz}$

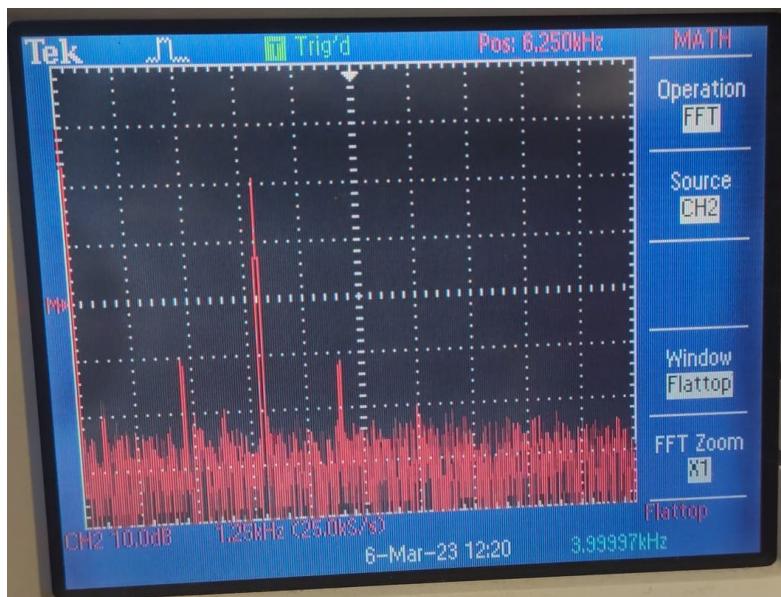


Figura 2.11: FFT do sinal de saída sem interpolação a $f_0 = 4\text{kHz}$

Por comparação entre as figuras 2.10 e 2.11 é possível observar que a interpolação linear não altera o espetro do sinal de saída para a frequência $f_0 = 4\text{kHz}$.

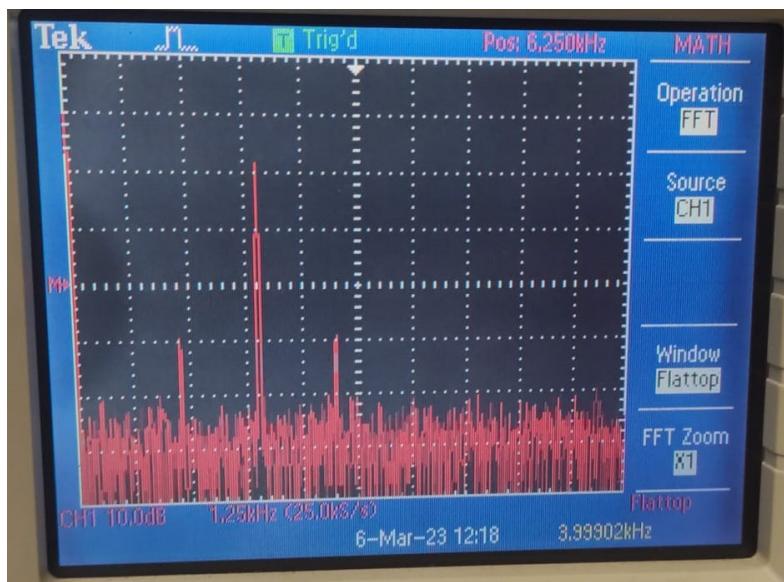


Figura 2.12: FFT do sinal de saída interpolado a $f_0 = 3.9991\text{kHz}$

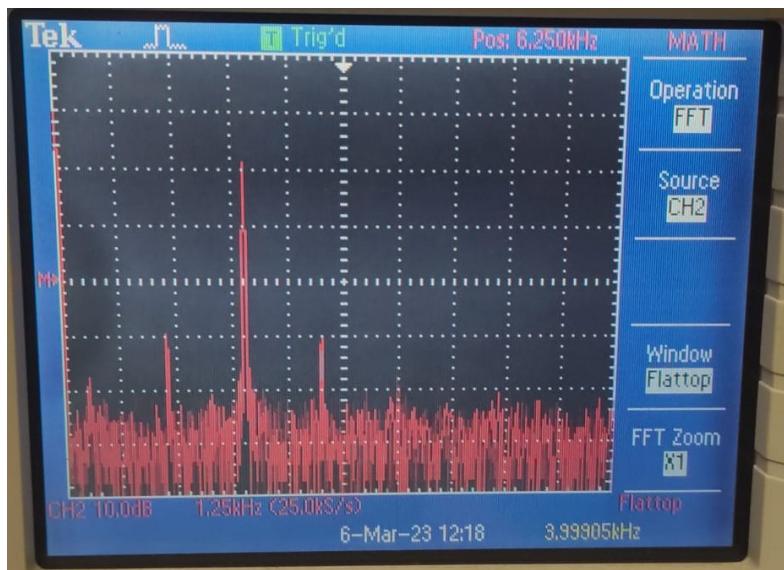


Figura 2.13: FFT do sinal de saída sem interpolação a $f_0 = 3.9991\text{kHz}$

Por outro lado, comparando as figuras 2.12 e 2.13, apesar de à primeira vista o espetro não se diferenciar nas duas imagens, analisando mais ao detalhe é possível verificar que quando se realiza a interpolação linear para $f_0 = 3.9991\text{kHz}$, elimina-se o chamado ruído de fase, que aparece junto à harmônica fundamental do sinal.

2.9 Questão 9

Nesta questão é pedido para justificar o porquê da equação presente na figura 2.9 pode ser representada no formato Q_{15} .

Primeiramente sabe-se que y_1 , y_2 e f estão no formato Q_{15} . De seguida, sabe-se também que y_1 e y_2 irão corresponder a valores sucessivos da LUT, a diferença entre estes valores será sempre menor que 1 (uma vez que a LUT indexa valores para uma sinusoide que está contida no intervalo $[-1;1]$). Ao multiplicar essa diferença por um valor que está ele também representado em Q_{15} e é também menor que 1, este resultado pode ser representado em Q_{15} . Por fim, ao somar um número representado em Q_{15} com outro no mesmo formato, de forma genérica, pode-se afirmar que o resultado estaria e só poderia ser representado, no melhor dos casos, no formato Q_{14} .

Neste caso iremos abordar dois cenários limites para facilitar a justificação o facto do resultado poder ser representado no formato Q_{15} .

Caso máximo e mínimo

Neste cenário iremos considerar que temos $y_1 = 1$ (que corresponde ao valor máximo que a sinusoide pode assumir). Inicialmente, poderá aparentar que o resultado da soma será maior que 1, no entanto, neste caso $f = 0$ uma vez que 1 é um número inteiro e, por isso, o resultado daria 1 (que está dentro do contradomínio da sinusoide $[-1;1]$). Para além disso, o resultado da soma seria negativo uma vez que y_2 seria menor que y_1 (uma vez que y_1 corresponde ao máximo da função) e, assim sendo, o resultado soma seria também menor que 1, podendo por isso ser representado no formato Q_{15} . Caso $y_1 = -1$ a situação é bastante semelhante, sendo que neste caso o valor resultante da interpolação é -1.

Caso próximo do limite

Neste segundo caso iremos considerar o valor imediatamente antes do máximo (ou seja 32609 que em Q_{15} representa 0.9951477051). Tendo este valor como y_1 é fácil de perceber que $y_2 - y_1 \approx 0.005$ e, uma vez que f é ele também menor que 1 o resultado dessa diferença, ao multiplicar por f ficaria ainda mais pequeno e, por isso, o resultado final nunca será maior que o 1. O mesmo raciocínio se aplica para o valor próximo do mínimo absoluto da função.