



Experimento 00

Laboratório de Princípios de Comunicação

Autoria	Matrícula
Pedro Henrique Dornelas Almeida	18/0108140

Engenharia de Redes de Comunicação
Universidade de Brasília

09 de Fevereiro de 2021

1 Introdução

O objetivo do experimento é a ambientação e primeiros projetos sendo implementados no *software* GNU Radio Companion, que é utilizado para a simulação de um ambiente experimental para tratar com sinais. Para isto realizaremos a implementação e visualização de sinais gerados, tanto no domínio do tempo, tanto no domínio da frequência, ambos objetos de estudo do curso.

Versão do GNU Radio Companion: 3.8.1.0 (Python 3.8.5).

2 Desenvolvimento

2.1 AR 01

O objetivo da atividade é explorar as funções de um osciloscópio, bem como gerar sinais em um gerador e analisá-los.

Faremos isso gerando 2 entradas, de forma a termos os 2 sinais simultaneamente sendo mostrados no osciloscópio. Usaremos uma senóide e uma onda quadrada, ambos de frequência 1kHz e amplitude de 1V.

2.1.1 Passos

Citarei aqui os passos mais importantes para que a análise possa ser feita de forma coerente, não citarei os passos já feitos no Roteiro do experimento, visto que eles são a base para o que faremos aqui:

1. Criar 2 geradores de sinais, um para senóide e um para onda quadrada, ambos de 1kHz e amplitude de 1V;
 - (a) Selecionar na área de blocos(AB): Waveform Generators → Signal Source;
 - (b) Configurar ambos de acordo com as imagens abaixo:

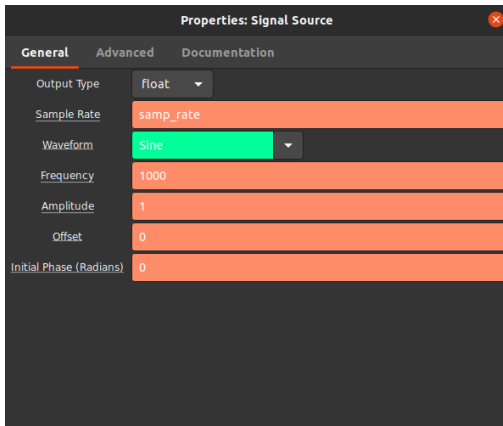


Figure 1: Gerador Senóide

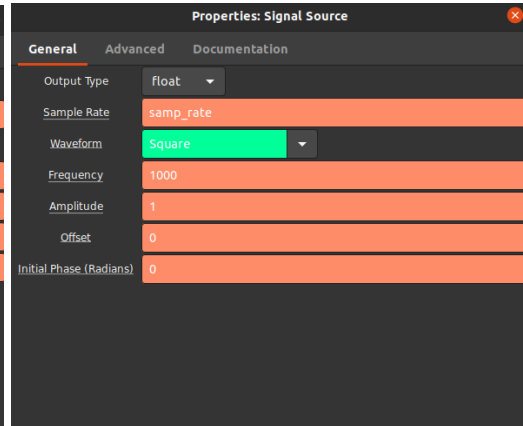


Figure 2: Gerador Onda Quadrada

2. Criar um único osciloscópio: AB → Instrumentation → QT → QT Time Sink;
3. Configurá-lo de acordo com a foto abaixo:

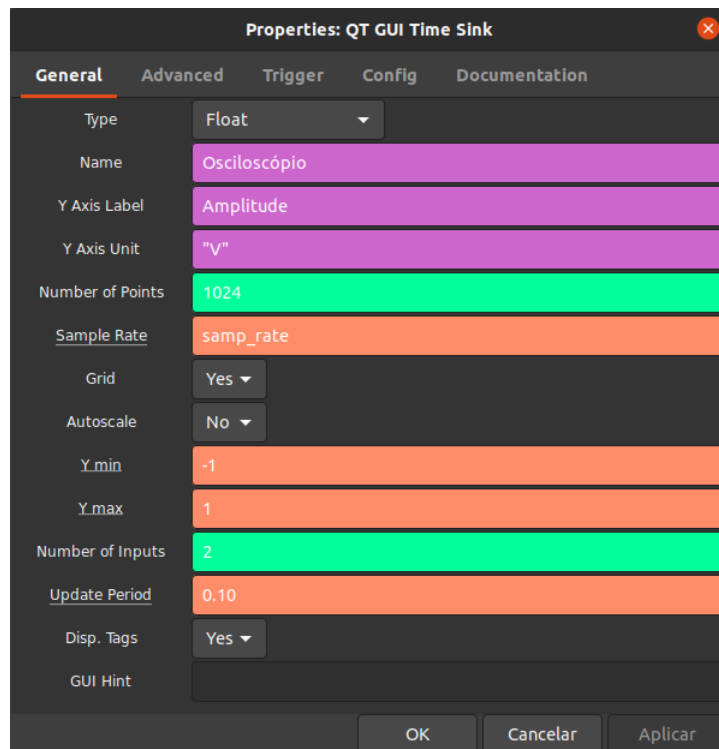


Figure 3: Osciloscópio Configurações

Aqui é importante explicar as funções principais que utilizaremos em um osciloscópio

- (a) Podemos alterar o campo Name para que ele apareça no nosso gráfico.
 - (b) O campo Y Axis Label: é importante pois podemos nomear o eixo y que aparece no nosso gráfico, e no campo abaixo Y Axis Unit podemos colocar a unidade em que esse eixo está. Ambas são informações muito necessárias a um gráfico.
 - (c) Number of points é o número de pontos que são mostrados pelo osciloscópio em um gráfico dentro de um intervalo de tempo.
 - (d) Sample Rate é o campo em que podemos alterar a taxa de amostragem do nosso osciloscópio.
 - (e) O campo grid é importante deixarmos habilitado pra facilitar a visualização dos gráficos, ele mostra linhas nos gráficos em relação aos valores dos eixos.
 - (f) Y min e Y max são os limites do gráfico que vem por padrão na visualização, eles são alterados a medida que usamos o scroll no gráfico.
 - (g) O campo Number of Inputs foi e será muito importante para as atividades, pois podemos controlar a quantidade de entradas aceitas e mostradas pelo osciloscópio.
 - (h) O último campo e não menos importante, GUI Hint, serve para organizarmos como o osciloscópio será visto quando entrarmos na execução dos projetos, ele é feito por uma matriz (linha_inic, coluna_inic, qtd_linhas, qtd_colunas), assim é organizado visualmente.
4. Criar 2 *Throttles* por boas práticas que vamos adotar futuramente para utilizarmos, ele serve para não sobrecarregar a CPU toda quando tivermos de implementar circuitos pesados em poder de processamento, logo ele ajuda a controlar o uso da máquina.
- (a) AB \rightarrow Misc \rightarrow Throttle
 - (b) Alterar em suas configurações para variável do tipo *float*, pois é como estamos utilizando todas as variáveis do experimento.

Dessa maneira, teremos todos os nossos componentes prontos para gerar os gráficos para a nossa análise, basta ligar os componentes como a área de trabalho abaixo:

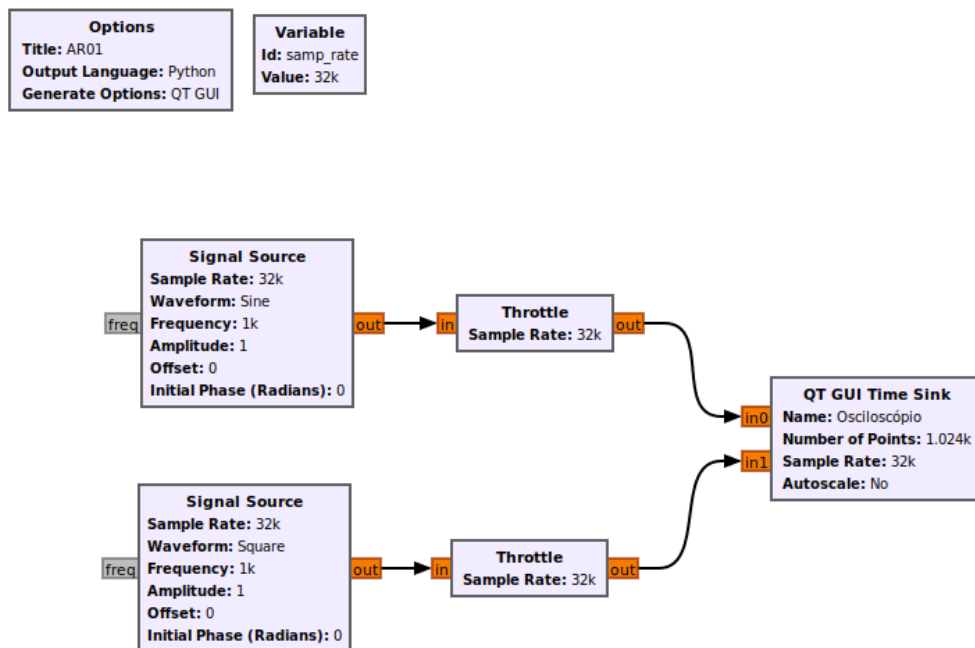


Figure 4: Área de trabalho AR01

Após isso, podemos rodar a aplicação e veremos o gráfico da figura 5, abaixo, em que iremos analisar os resultados obtidos.

2.1.2 Análise

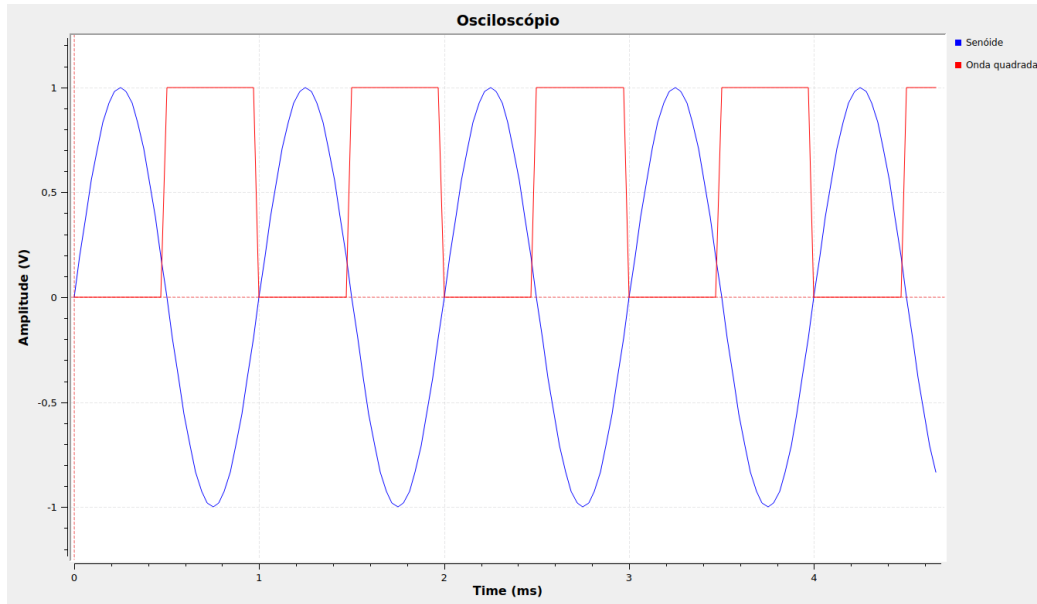


Figure 5: Osciloscópio AR01

Para nossa análise é importante citar que utilizamos todas as variáveis e sinais como tipo *float*, portanto, todos as entradas e saídas dos componentes devem ser configurados dessa maneira.

Podemos ver que geramos uma onda senoidal(curva azul) de fato e uma onda quadrada(curva vermelha), ambas de frequência 1kHz e 1V de amplitude. Conseguimos ver uma pequena diferença nos gráficos no ponto em que a onda quadrada tem seu pulso de subida, em que vemos uma diferença em relação a quando a senoide passa ao lado negativo. O motivo dessa assincronia é a taxa de amostragem e a quantidade de pontos que existem no gráfico, então é uma assincronia apenas aparente, visto a irregularidade entre taxa de amostragem e frequência das ondas.

2.2 AR 02

O objetivo desse exercício é aplicar os conhecimentos desenvolvidos em exercícios anteriores, porém agora, utilizando mais algumas ferramentas que permitem a manipulação dos sinais durante o experimento.

Devemos criar uma cossenóide com frequência e amplitude configuráveis. Usaremos uma taxa de amostragem de 32ksps(*kilosamples per second*, ou kiloamostras por segundo).

Devemos criar também um controle para o *offset* do gerador de forma de onda, faixa entre $[-2V, +2V]$ e verificar o sinal no domínio do tempo e da frequência.

2.2.1 Passos

Citarei aqui os passos mais importantes para que a análise possa ser feita de forma coerente, não citarei os passos já feitos no Roteiro do experimento, visto que eles são a base para o que faremos aqui:

1. Continuaremos a utilizar variáveis do tipo *float*, se atentar a deixar as saídas e entradas de componentes desse tipo.
2. Criar um controlador de frequência, um de amplitude e um de *offset*:
GUI Widgets → QT GUI Range; Configurá-los como mostra o roteiro;
(a) Para configurar o *offset*, fazer como na imagem abaixo:

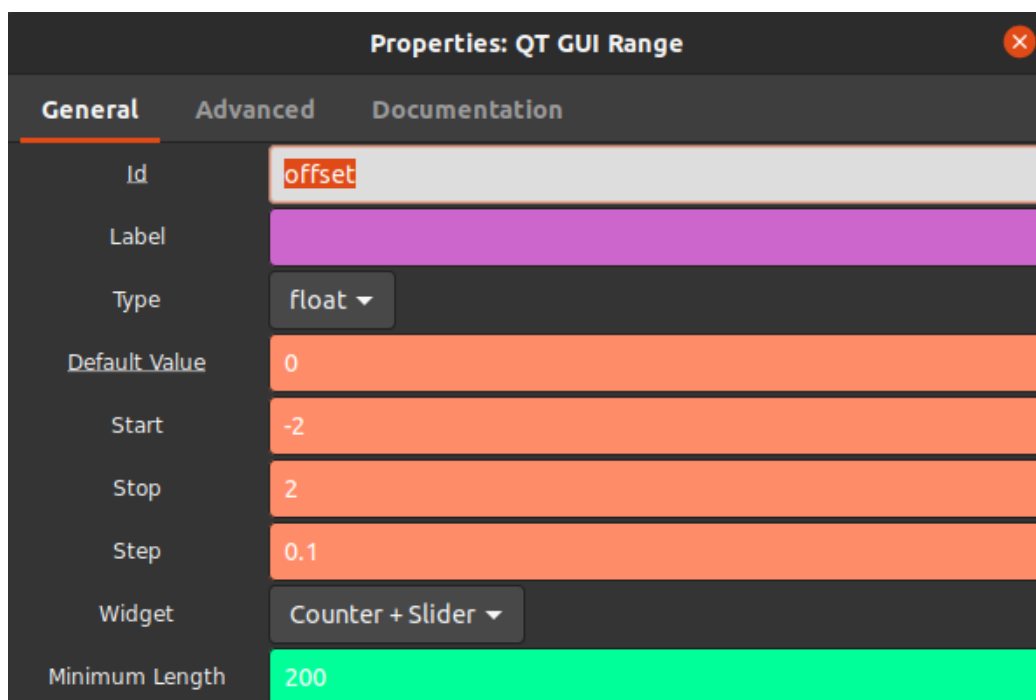


Figure 6: *offset*

3. Criar um gerador de sinal: Waveform Generators → Signal Source;

- (a) Configurá-lo para gerar um cosseno e em suas variáveis colocar as variáveis criadas no passo anterior para que funcione corretamente quando gerarmos o gráfico.
4. Criar um Throttle: Misc → Throttle;
 5. Criar um Osciloscópio seguindo os passos da atividade anterior.
 6. Criar um analisador do espectro de frequência: Instrumentation → QT → QT GUI Frequency Sink; Apenas alterar a entrada para float;

Após seguir estes passos, teremos o nosso ambiente quase pronto para realizar o experimento, devemos então ligar os componentes e ter todos os componentes abaixo:

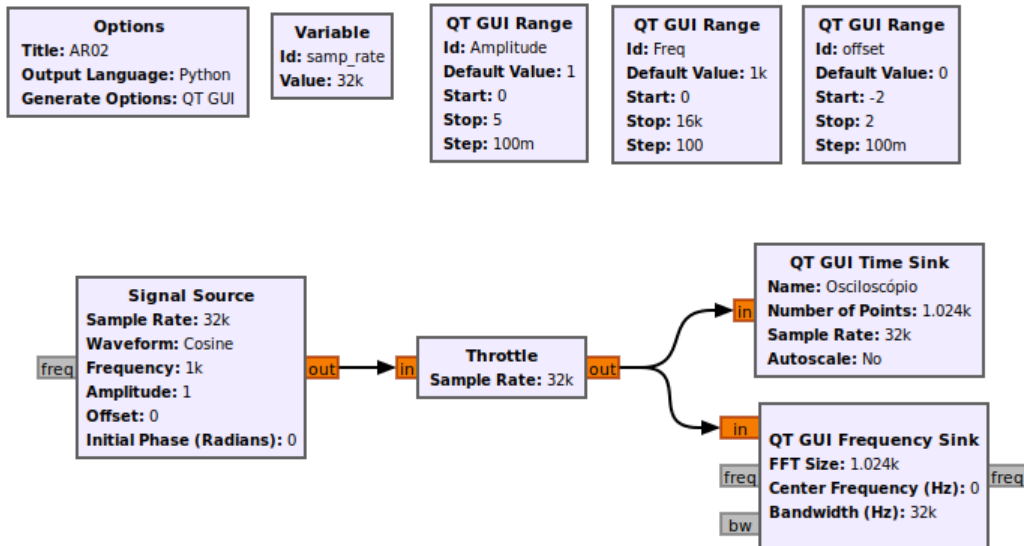


Figure 7: Área de trabalho AR02

Foi usado também um esquema de visualização dividido em matrizes que o GNU Radio permite utilizar para dividir melhor os gráficos e controles de variáveis utilizados.

2.2.2 Análise

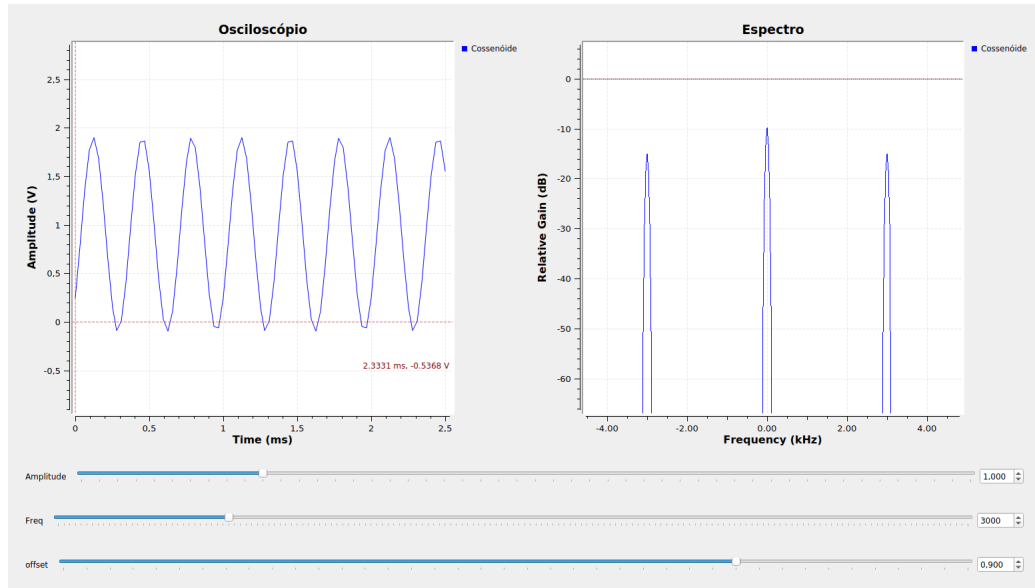


Figure 8: Análise AR02

Para a análise alteramos as variáveis de frequência e de *offset* para obtermos um resultado ainda não visto. Lembrando que usamos uma cossenóide para analisá-la. Podemos ver que no analisador de espectro a frequência vista na cossenóide é exatamente a colocada no controlador de frequência, sendo esta 3kHz, o que comprova que o nosso controlador e gerador de sinais estão funcionando corretamente.

Outro ponto pertinente para análise é o fato de que quando alteramos o *offset* da onda para um valor diferente de 0V, podemos observar no analisador do espectro que apareceu uma frequência em 0kHz, acredito que para mudar o *offset* o sinal é somado no domínio do tempo por um sinal constante com a amplitude do *offset*, exemplo:

$$2 + \sin 60^\circ$$

assim, no domínio da frequência aparece um impulso na frequência 0Hz, o que explica o gráfico.

2.3 AR 03

O objetivo dessa atividade é comparar os diferentes valores de Densidade Espectral de Potência(DEP), vistos em 3 diferentes sinais, todos com am-

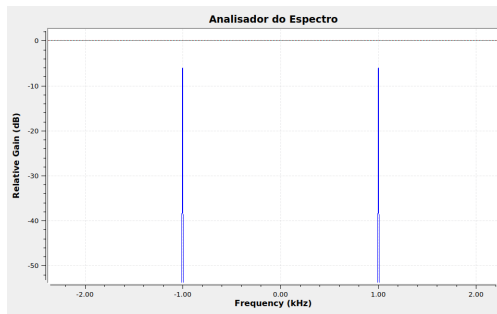
plitude de 1V, frequência de 1kHz, e o *offset* variando entre 3 valores: 0V, 0,5V, 1V.

2.3.1 Passos

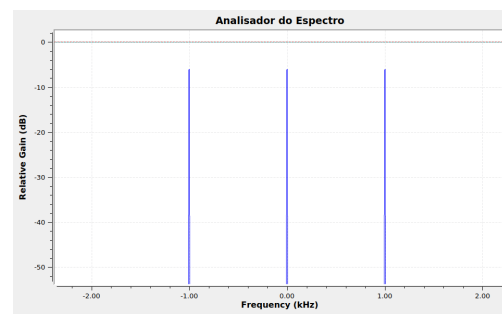
Usaremos a mesma configuração da atividade anterior, podemos ver na Fig.7, porém, desabilitamos o osciloscópio para analisar somente o espectro da frequência.

Outra mudança que realizamos foi retirar o janelamento da FFT: deixar a opção "*None*". Assim, acredito ter o ambiente pronto para realizarmos a análise.

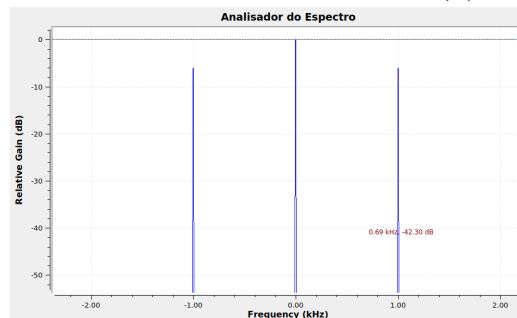
2.3.2 Análise



(a) *offset*=0V



(b) *offset*=0.5V



(c) *offset*=1V

Figure 9

$offset(V)$	-1kHz	0kHz	1kHz
0V	-6dB	-	-6dB
0,5V	-6dB	-6dB	-6dB
1V	-6dB	0dB	-6dB

Table 1: Densidade Espectral de Potência

Analisando os dados coletados na tabela 1, percebemos que existe uma certa regularidade. Com o aumento do *offset*, acontece também um aumento no DEP(Densidade Espectral de Potência), o que faz sentido, pois é como somar um sinal com uma amplitude cada vez maior.

2.4 AR 04

O objetivo dessa atividade é aprender usar os operadores matemáticos e explicar o motivo do gráfico ter se alterado e explicar o porque ele se altera ao fazer operações com sinais.

Para isso, usaremos dois geradores de sinais(cossenóides), ambos com frequência ajustável entre [0Hz, 20kHz], amplitude fixa em 1V, *offset* em 0V e taxa de amostragem padrão em 32ksps. Manteremos a configuração do analisador da atividade remota 3, como pede o roteiro.

Para usar os operadores matemáticos, devemos implementar o diagrama de blocos da figura:

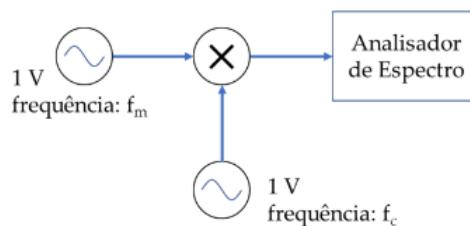


Figure 10: Diagrama de Blocos

2.4.1 Passos

Além dos passos de configurações que já fizemos anteriormente, como criar o gerador de sinais e o analisador de frequência e usando as configurações já feitas anteriormente para esses dispositivos, passamos as partes que não fizemos ainda:

1. Criar 2 controladores de frequência na faixa pedido, um para cada sinal;

2. Criar um multiplicador de sinais: Math Operators → Multiply ;
3. Alterar para *float* para ter compatibilidade com os outros componentes;

Após isso, espera-se ter a Área de Trabalho como a mostrada abaixo:

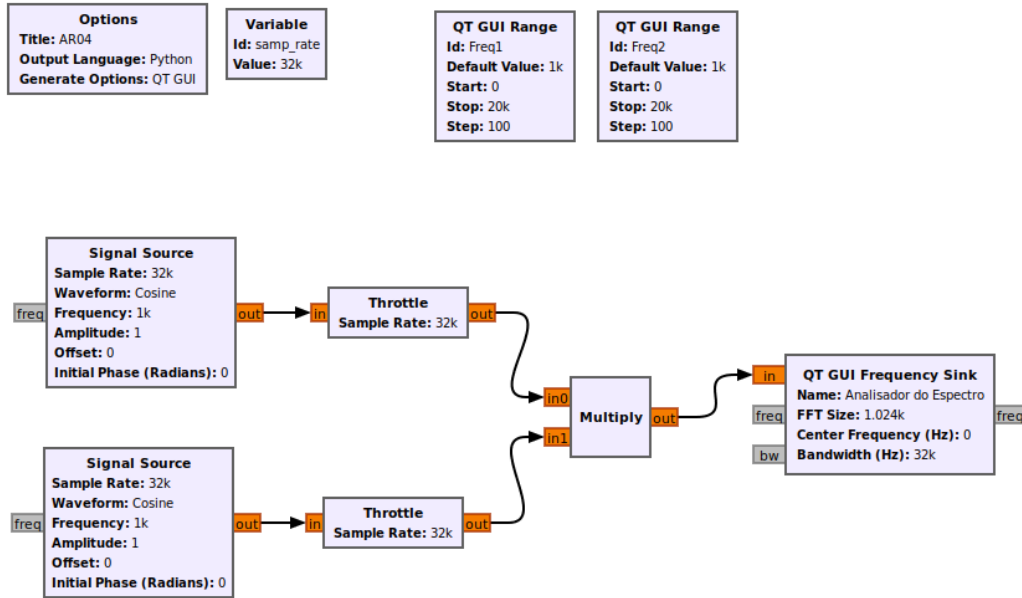


Figure 11: Área de Trabalho AR04

Feito isso, estamos prontos para executar o programa e analisar os gráficos gerados com os valores de frequência pedidos no roteiro.

2.4.2 Análise

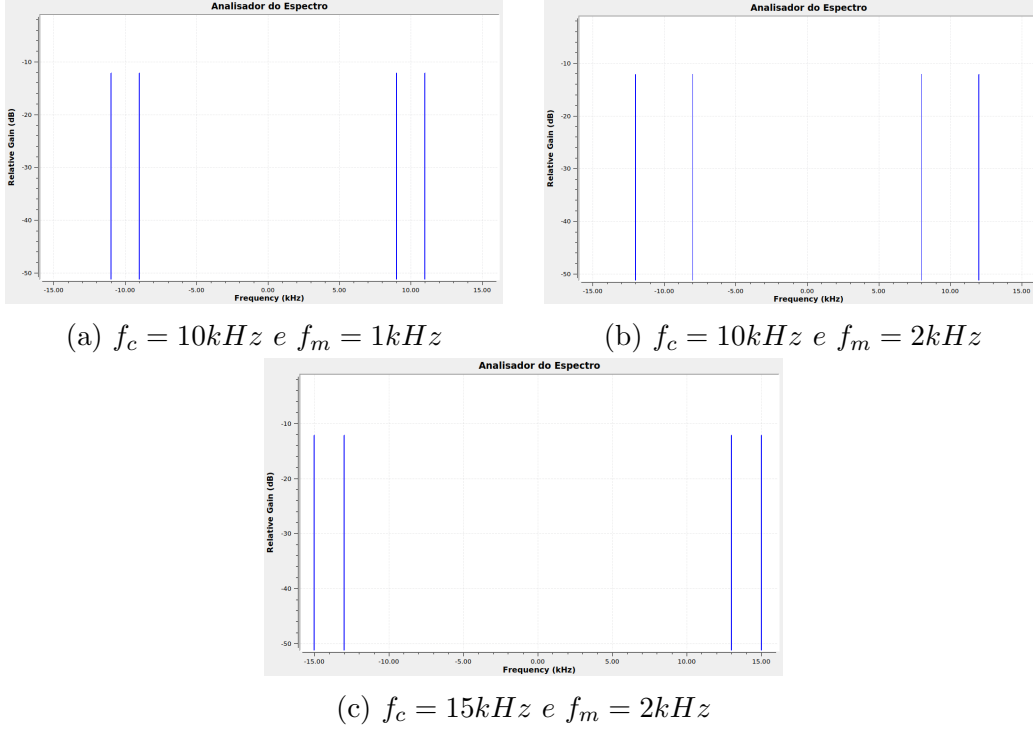


Figure 12

Notamos pelas imagens acima que o teorema de Nyquist se faz valer nos experimentos. Ele nos diz que a frequência da onda tem de valer menos da metade da taxa de amostragem, de forma que a frequência máxima permitida no experimento é:

$$f = \frac{\Omega}{2} = \frac{32kHz}{2} = 16kHz \quad (1)$$

Dessa maneira, como podemos ver na figura 12c, quando colocamos duas cossenóides sendo multiplicadas no domínio do tempo, elas serão convoluídas no domínio da frequência, assim, podemos transformar um cosseno em uma somatória de exponenciais, por meio da fórmula de Euler:

$$\cos x = \frac{e^{xj} + e^{-xj}}{2} \quad (2)$$

Dessa maneira, as frequências serão somadas e subtraídas no domínio da frequência, como vemos no analisador de espectro, e quando colocamos

$f_c = 15kHz$ e $f_m = 2kHz$, teremos $f_1 = 13kHz$ e $f_2 = 17kHz$, porém, como o teorema de Nyquist (equação 1) nos permite, nosso analisador pode mostrar somente até 16kHz, e temos uma espécie de *overflow* que acontece, então, ele mostra uma frequência aparente em 15kHz, provocada pela diferença $16kHz - 17kHz = -1kHz$, logo: $16kHz - 1kHz = 15kHz$. Isso está sendo mostrado na figura.

Já nas figuras 12a e 12b, o que ocorre é o esperado, as frequências que aparecem são a soma e subtração das frequências dos dois sinais:

$$f_1 = 9kHz \text{ e } f_2 = 11kHz \quad (3)$$

$$f_1 = 8kHz \text{ e } f_2 = 12kHz \quad (4)$$

3 Conclusão

Concluimos então o nosso experimento 00, de forma que conseguimos nos ambientar ao software, usar os componentes mais simples e assim praticamos para estarmos preparados para os próximos experimentos, que provavelmente serão mais complexos.

Conseguimos também aplicar os conceitos vistos na teoria na análise de sinais, de forma a entender os resultados práticos vistos no analisador de espectro e no osciloscópio, mesmo ainda não tendo visto a matéria na disciplina Princípios de Comunicação, pudemos ter uma base em disciplinas anteriores que nos possibilitou uma análise superficial no AR04 por exemplo.