

Tutorial

**Modulação QPSK no Simulink e
Geração de VHDL para FPGA**

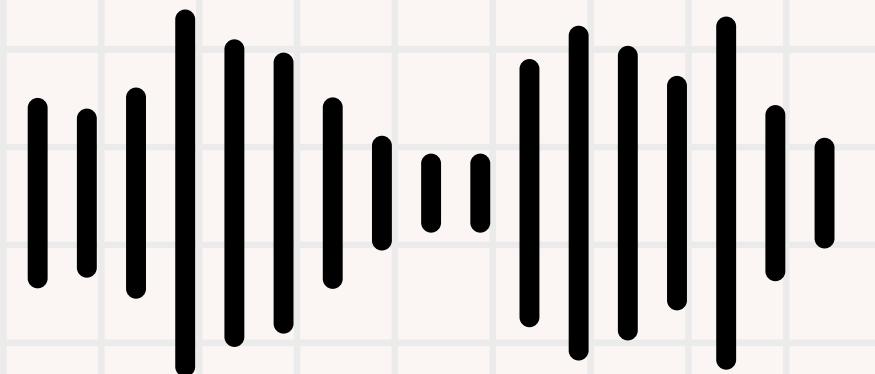
**Laboratórios Didáticos para Ensino de Sistemas de
Comunicação em FPGA**

Materiais

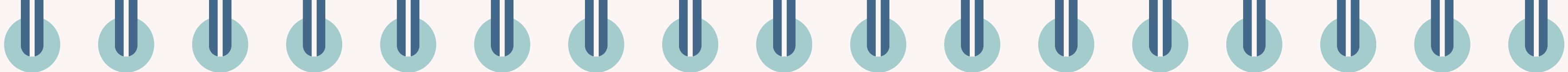
Pré-requisitos



MATLAB + Simulink
instalados



Conceito rápido da
QPSK



Introdução

Simulink

A Modulação QPSK (Quadrature Phase Shift Keying) é uma técnica eficiente que transmite 2 bits por símbolo, combinando as informações em sinais senoidais e cossenoidais. No ambiente Simulink, é possível modelar o modulador QPSK de forma intuitiva por meio de blocos. Isso permite validar o desempenho do sistema por simulação, observando a constelação de saída



Desenvolvimento



Passo a Passo



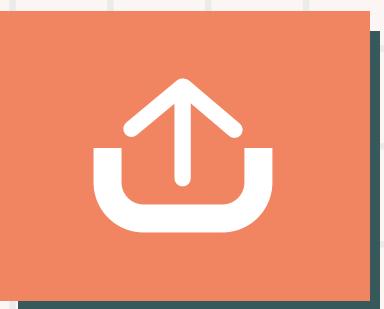


Desenvolvimento

Simulink

Arquivo do projeto

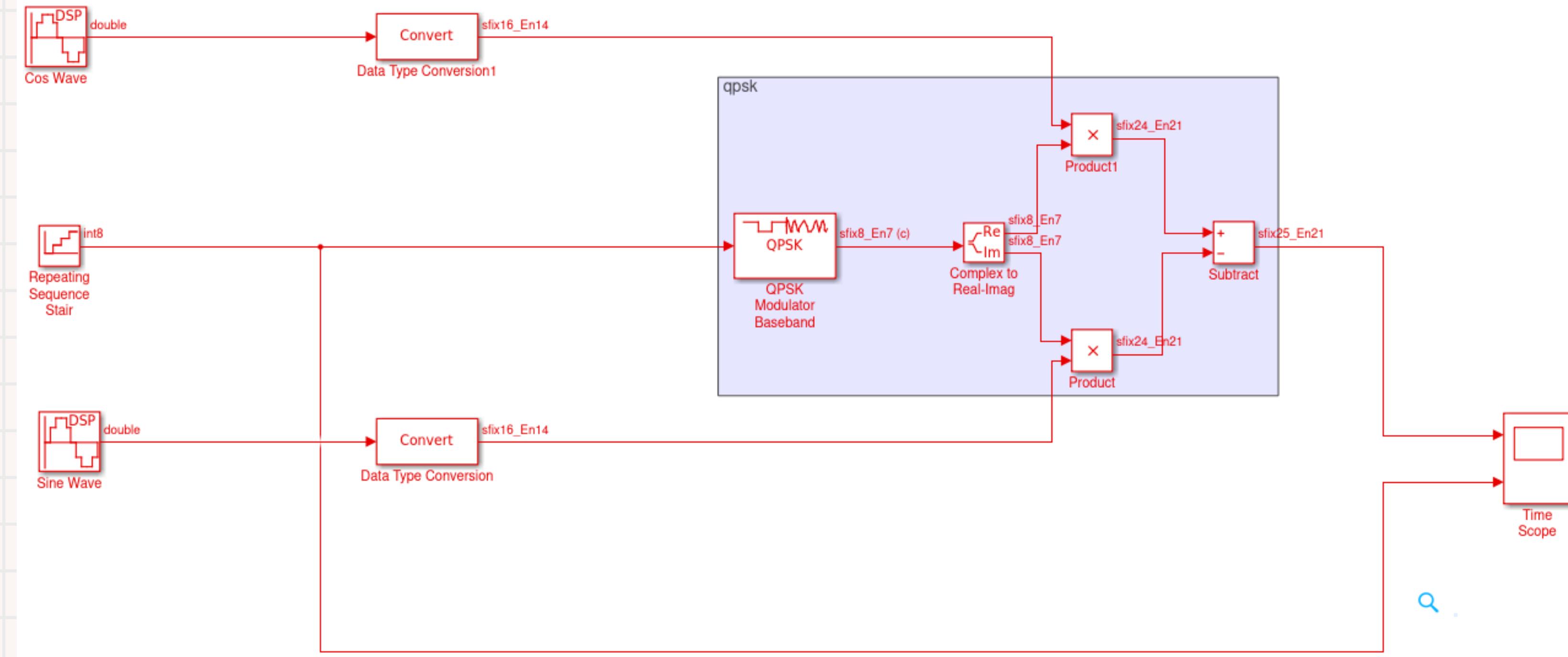
Para facilitar, é disponibilizado arquivo do modelo em Simulink no Github:



[Link Download](#)

Desenvolvimento

Modulação QPSK - Sistema



Desenvolvimento

Modulação QPSK - Sistema

No projeto você já encontra:

- Fonte de bits (Random Integer Generator) → Gerando números de 0 à 3
- Mapeamento QPSK (QAM Modulator Baseband com M=4)
- Separação I/Q (Complex to Real-Img)
- Sinais(para gerar seno/cosseno) → Como portadora
- Multiplicação e subtração → sinal QPSK modulado
- Blocos de conversão para Fixed-Point (compatível com HDL Coder)
- Time Scope para visualização



Desenvolvimento

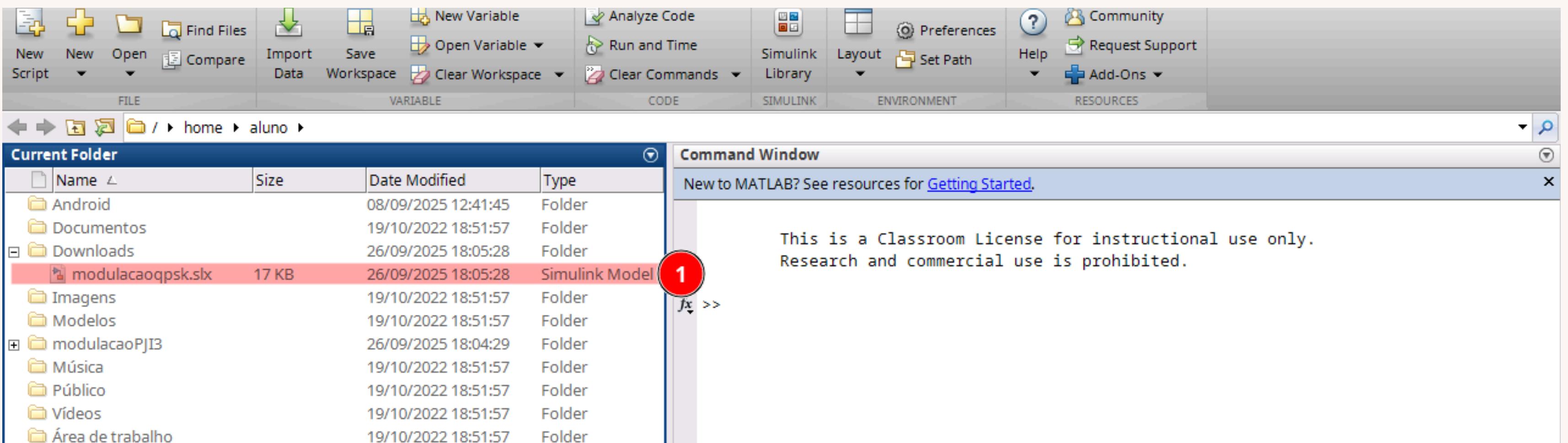
Modulação QPSK - Sistema



A seguir a demonstração de como usar o Simulink para geração automática de código VHDL...



Desenvolvimento



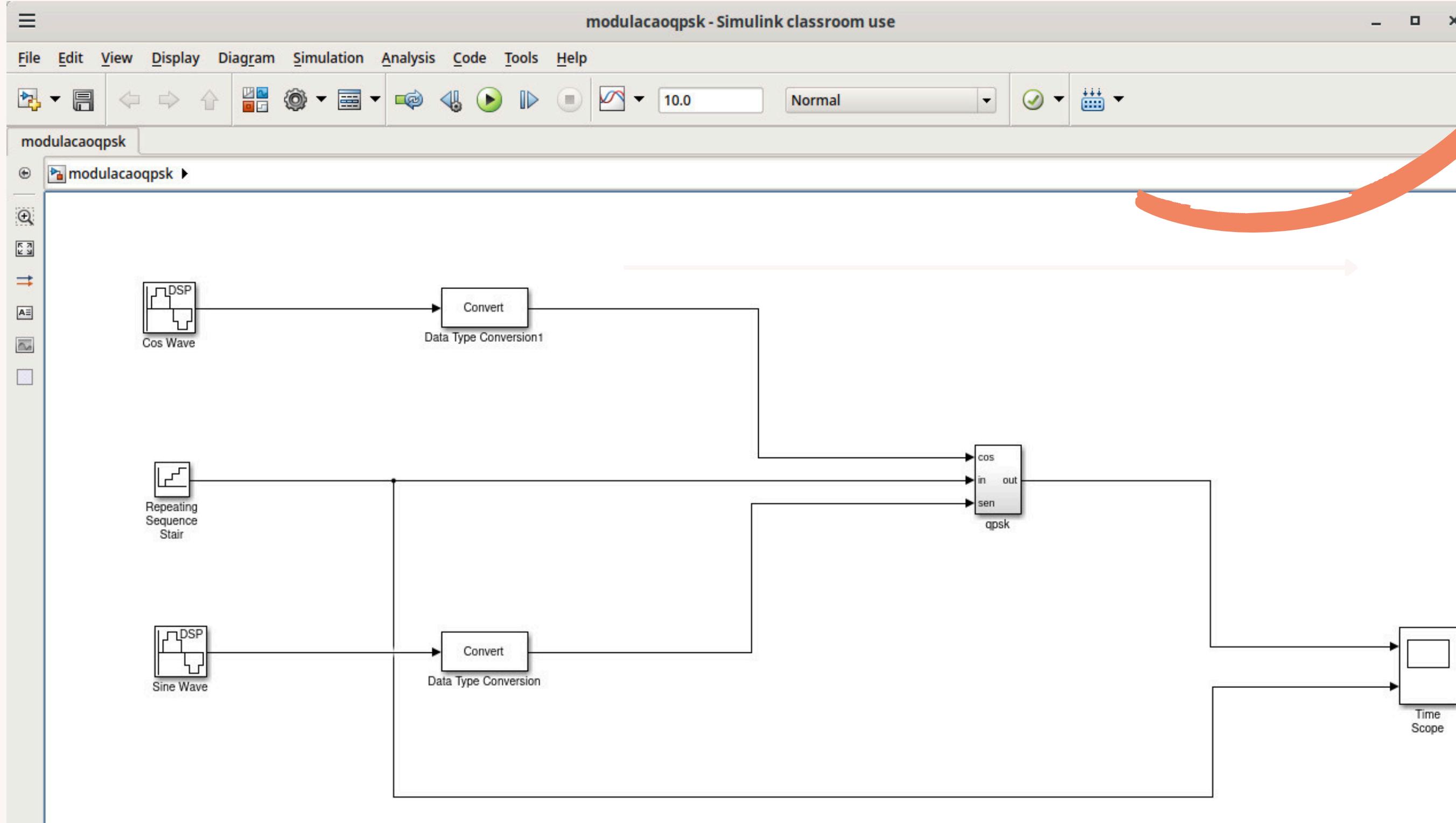
Abra o Matlab

Vá até ao diretório onde foi salvado o arquivo **modulacaoqpsk.slx**

1 - Dê um duplo click

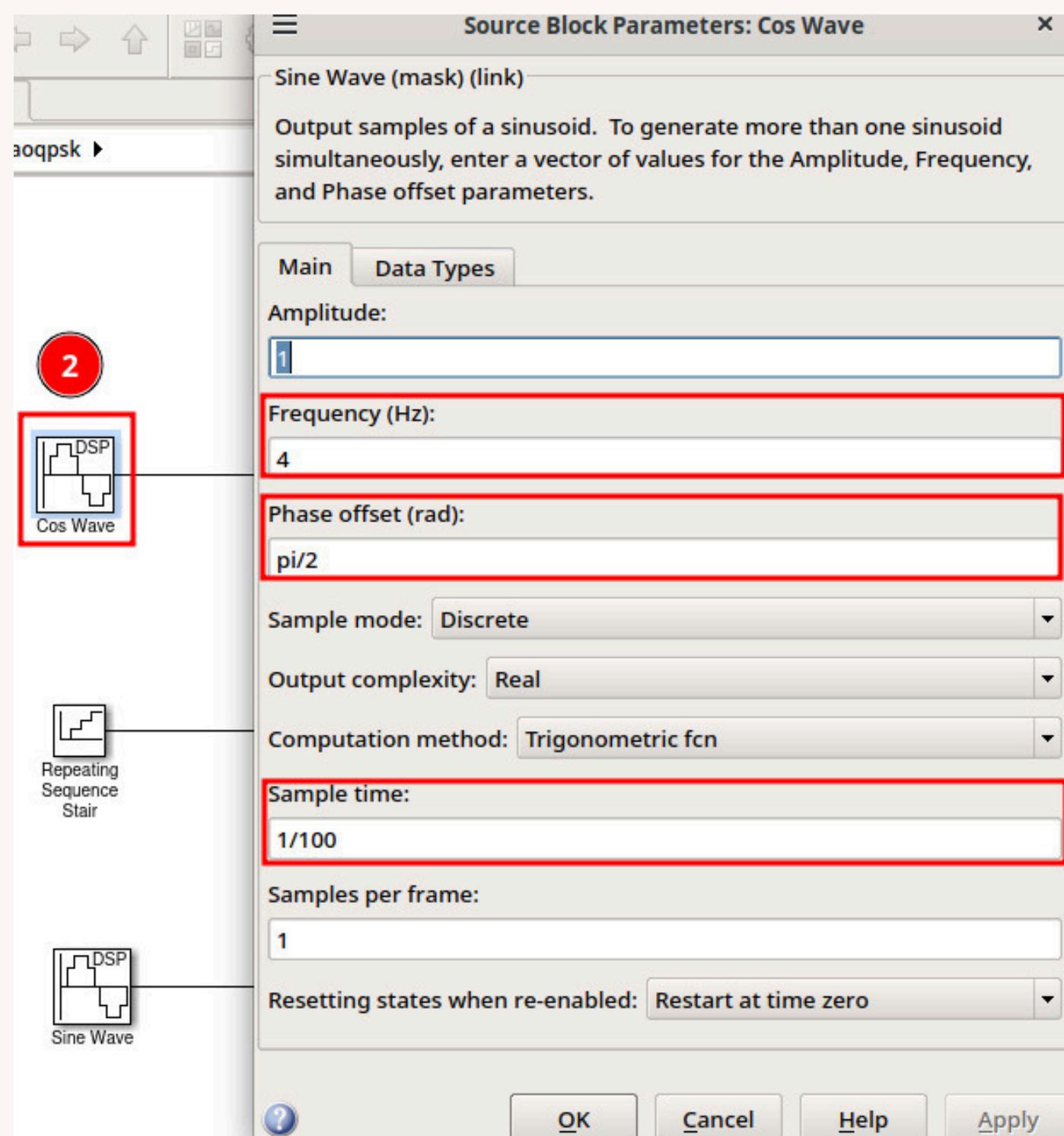
Desenvolvimento

modulacaoqpsk - Simulink classroom use



Abrirá essa aba
do simulink

Desenvolvimento



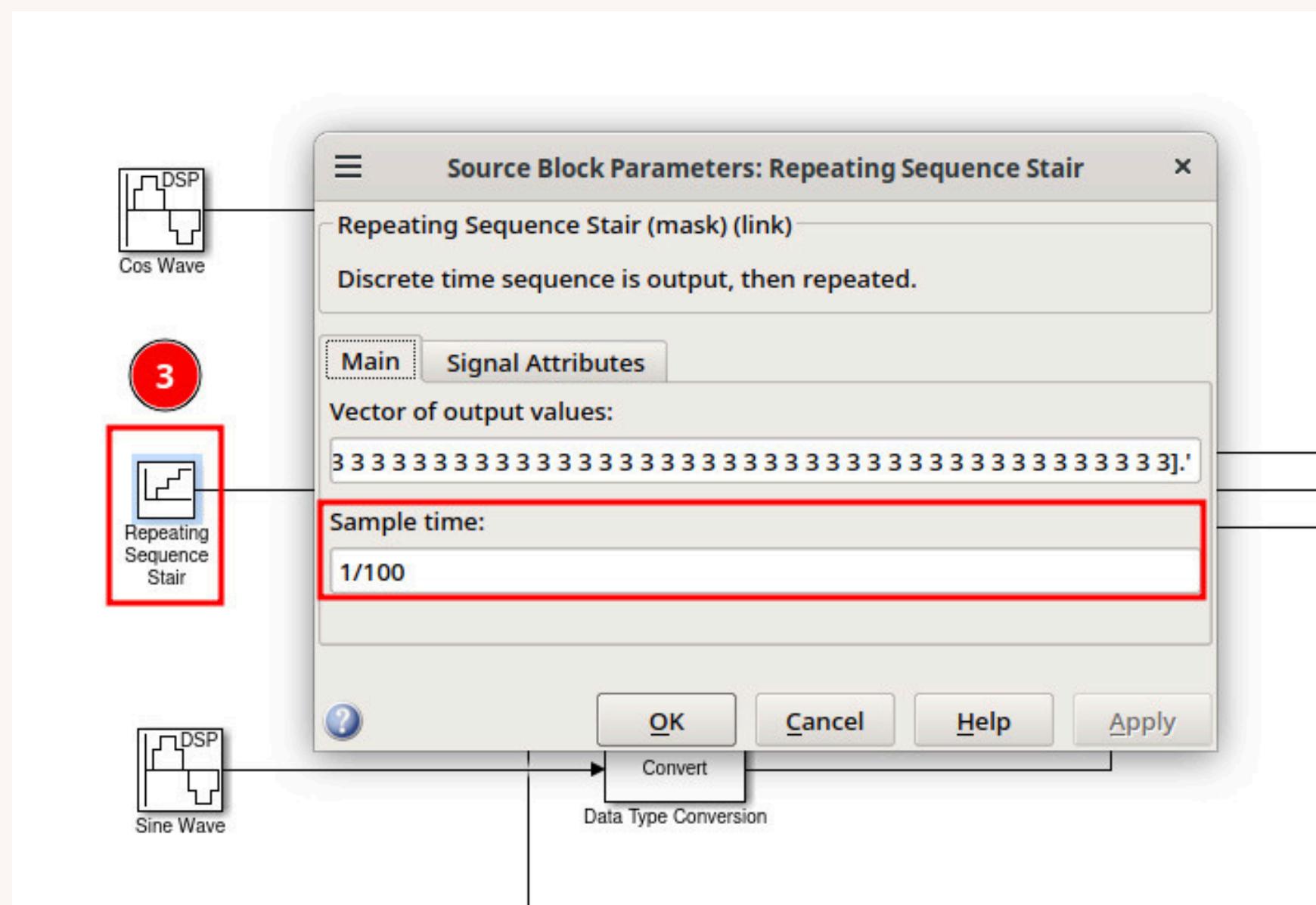
- Note que existe 3 entradas
Cosseno
Seno
Entrada aleatória de Θ a 3

2 - Cos Wave:

- **Frequency = 4:** Define a frequência da portadora. É um valor baixo e didático para que seja fácil de visualizar na simulação.
- **Phase offset = $\pi/2$:** Como não existe o bloco de cossenoide, uma onda senoidal é defasada em $\pi/2$ radianos (90°) é matematicamente igual a uma onda cossenoidal. $\cos(\theta) = \sin(\theta + \pi/2)$.
- **Sample time = 1/100:** Define a frequência de amostragem para 100 Hz.



Desenvolvimento

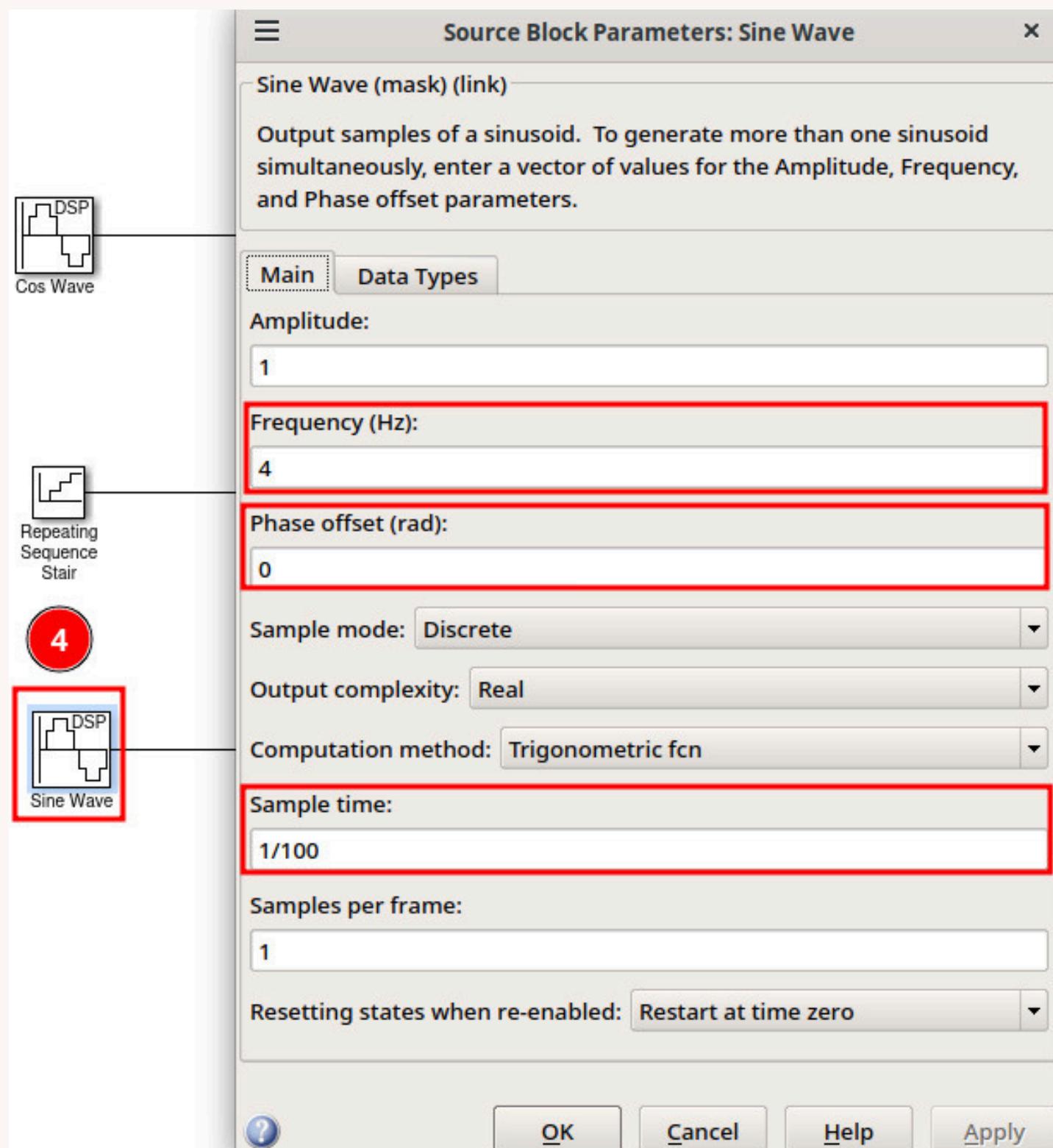


Todos os sinais devem ter o mesmo valor de amostragem: “sample time”!



3 - **Sample Time = 1/100**: Garante que a taxa de dados (símbolos) e a taxa das portadoras estejam perfeitamente alinhadas, permitindo que o modulador funcione corretamente.

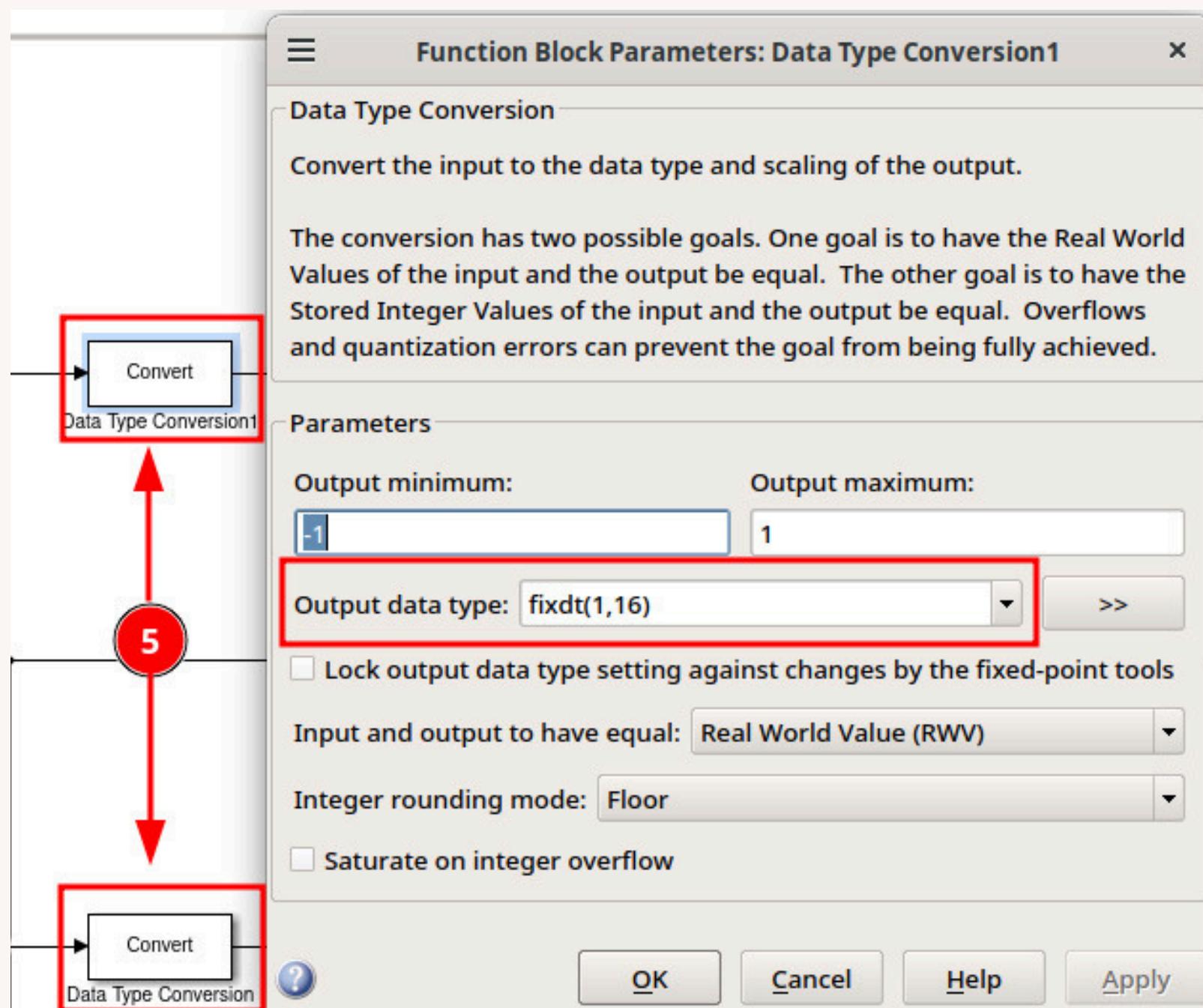
Desenvolvimento



4 - Sin Wave:

- **Frequency = 4:** A frequência da portadora deve ser idêntica à frequência da outra portadora (Cosseno) para que o sistema funcione corretamente.
- **Phase offset = 0:** Este valor mantém a função na sua forma senoidal pura ($\sin(\omega t)$).
- **Sample time = 1/100:** Obrigatório para sincronização!

Desenvolvimento

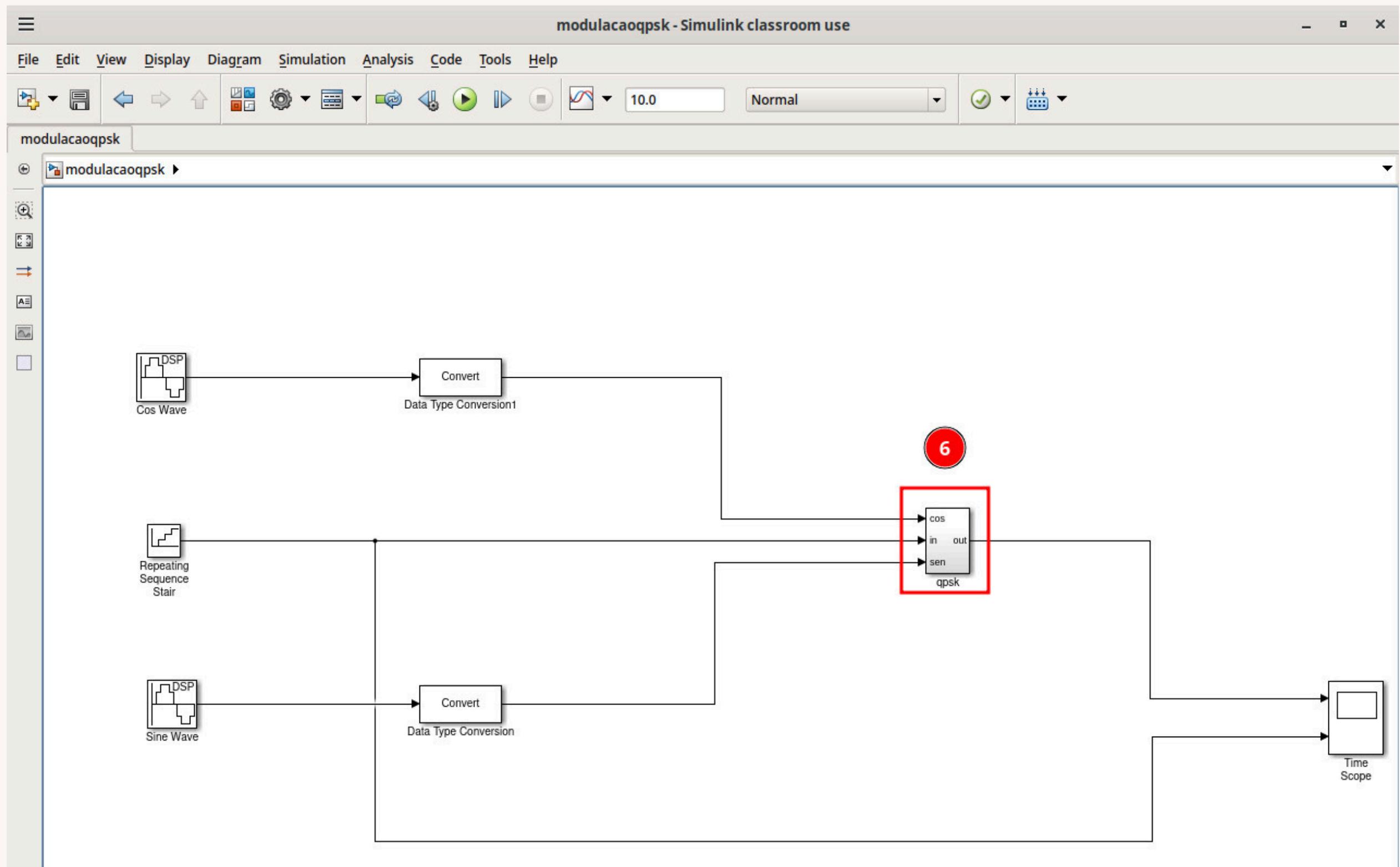


5 - Data Type Conversion

Compatibilidade de Blocos: Garantir que todos os blocos do sistema recebam o tipo de dado que eles esperam, evitando erros de simulação.

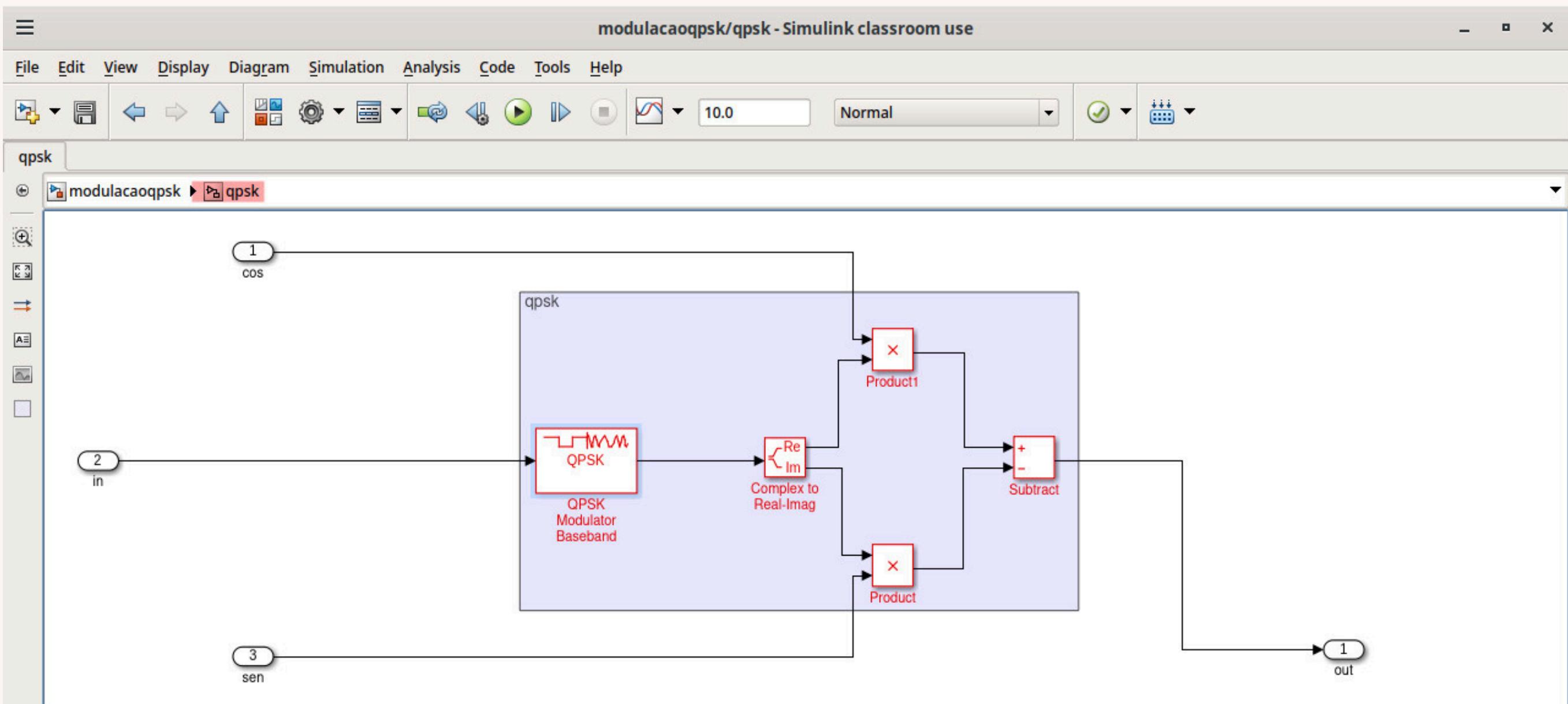
- **Output data type = fixdt(1, 16):** O sinal analógico de alta precisão (double) é arredondado e limitado para caber em apenas 16 bits.

Desenvolvimento



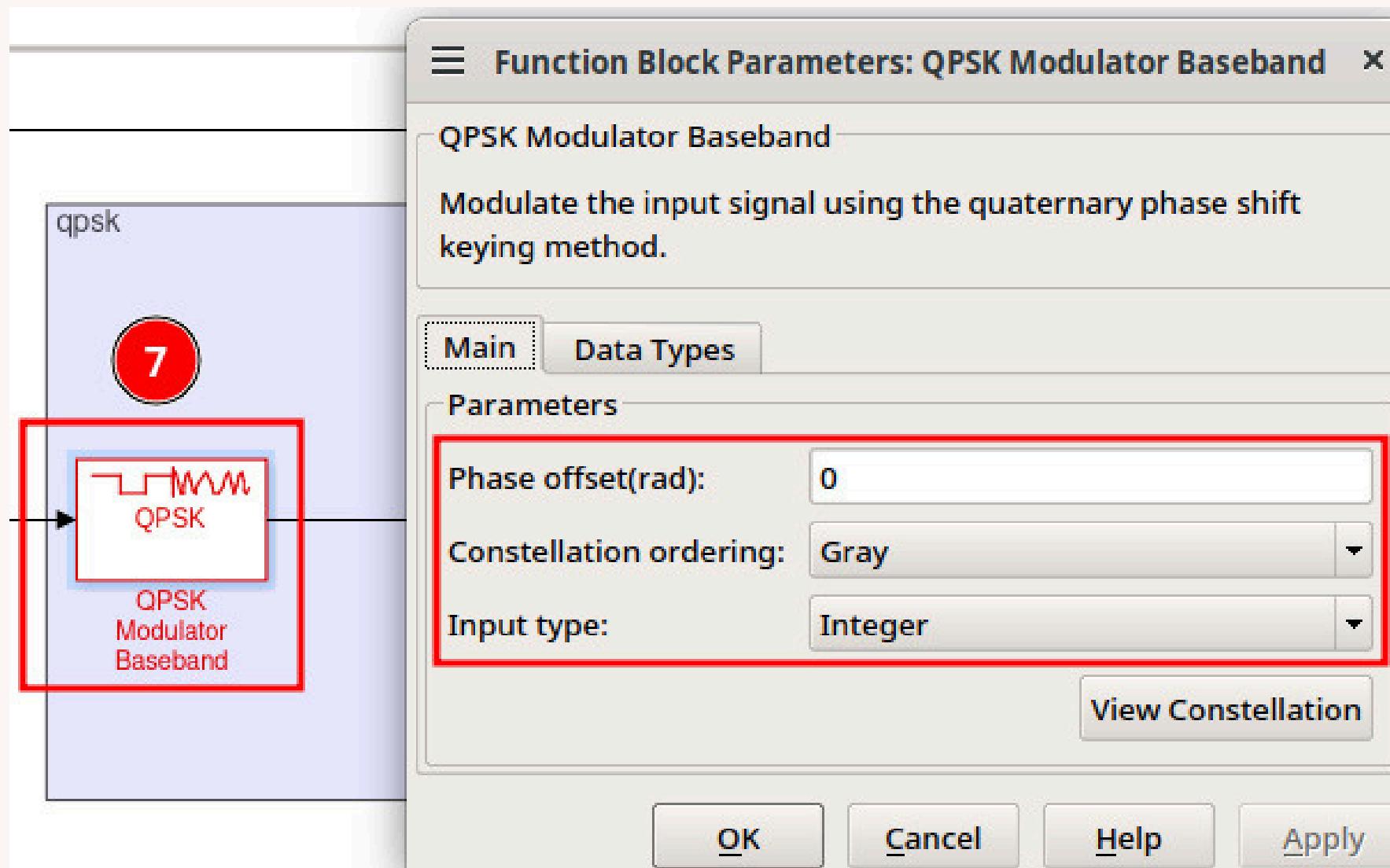
6 - Dando um
duplo click...

Desenvolvimento



Abrirá o
subsistema **qpsk**
(note que há
uma **hierarquia**)

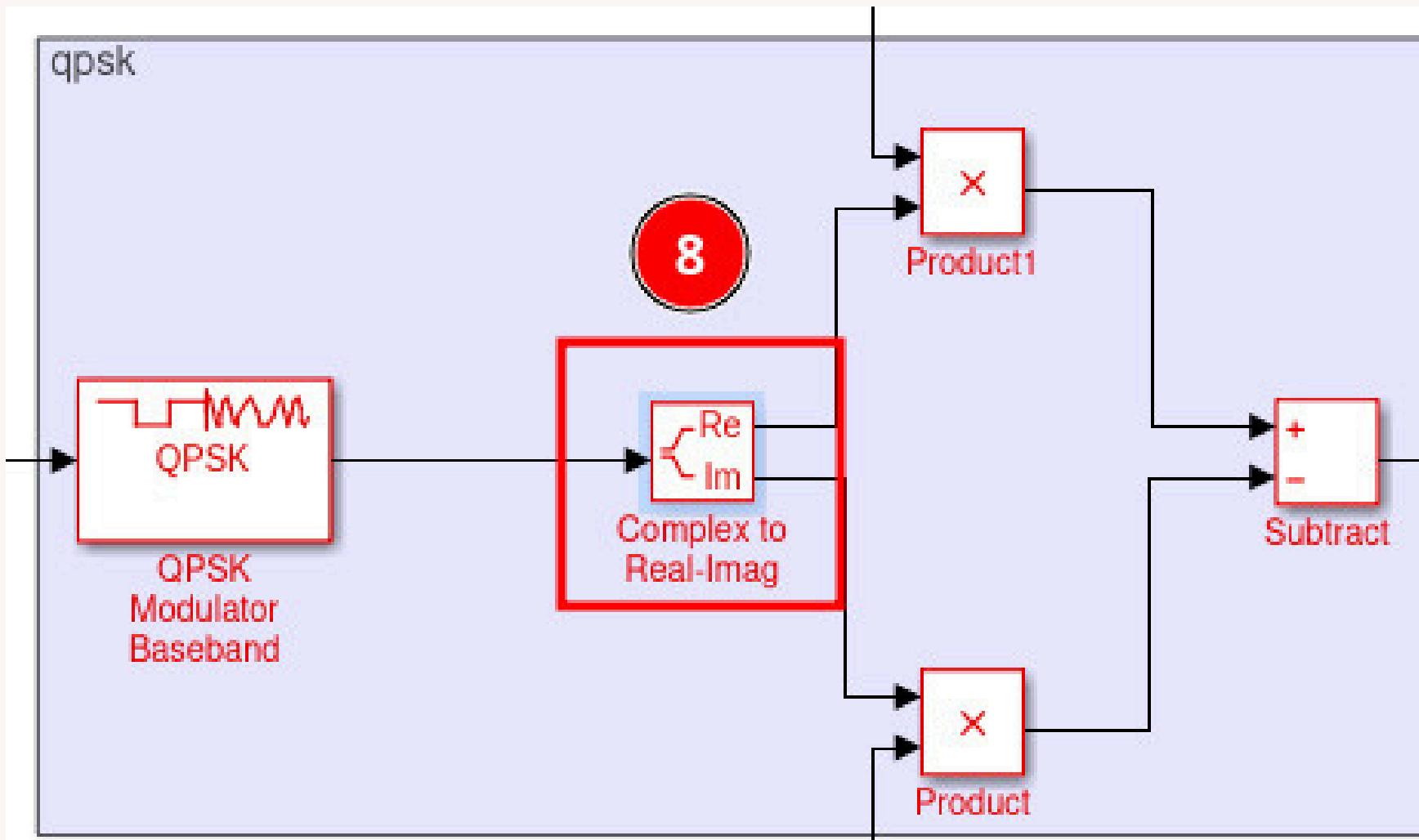
Desenvolvimento



7 - QPSK Modulator Baseband

- **Phase offset (rad) = Θ :** Define a fase inicial do primeiro ponto da constelação QPSK. Com 0° (zero radianos), os quatro símbolos serão mapeados nas fases de $0^\circ, 90^\circ, 180^\circ$ e 270° .
- **Constellation ordering = Gray:** Garante que símbolos adjacentes na constelação difiram em apenas um bit, reduzindo a chance de perder muitos dados em caso de ruído.
- **Input type = Integer:** Confirma que o bloco espera receber os dados como números inteiros (0, 1, 2, 3), que é a representação dos símbolos QPSK.

Desenvolvimento

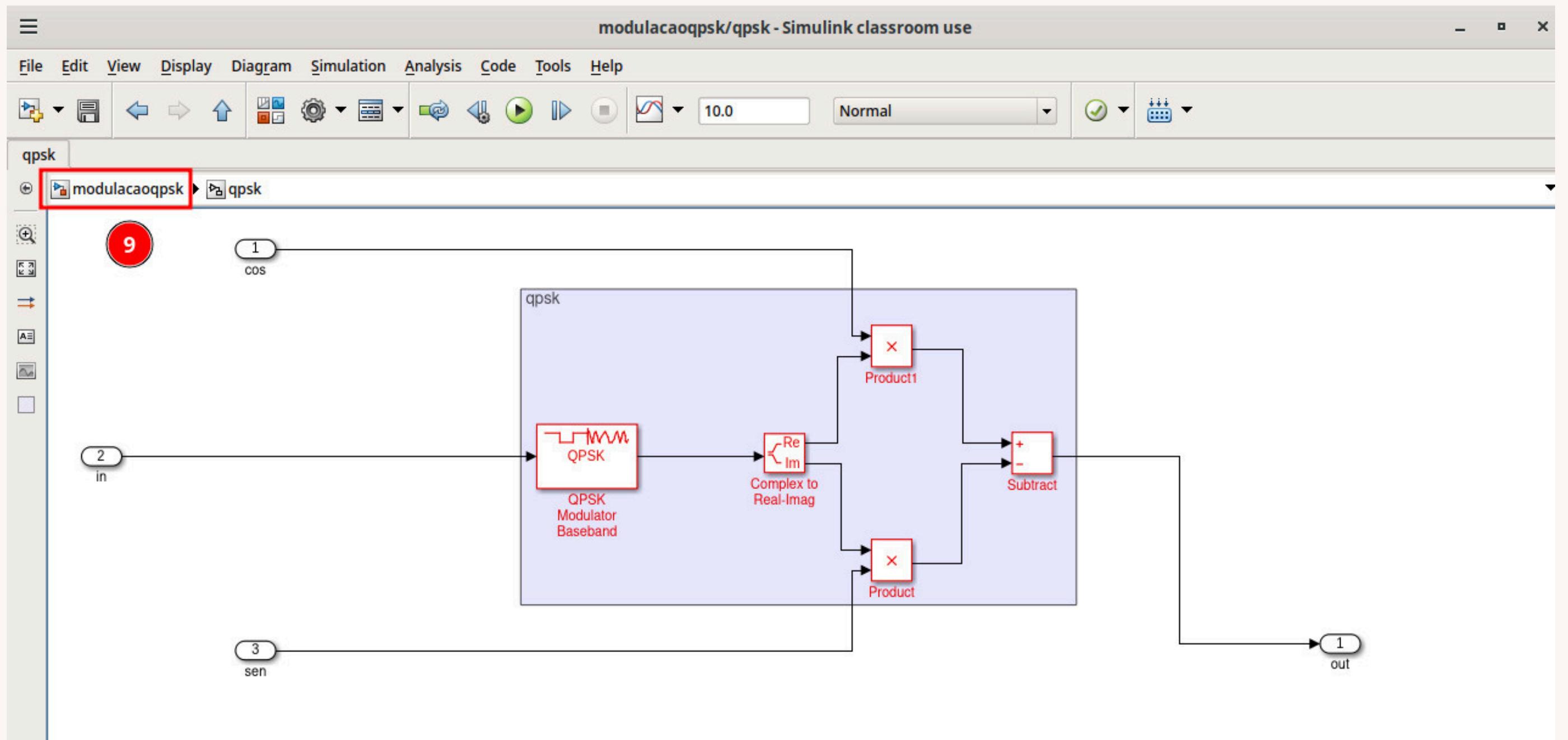


8 - Complex to Real-Imag

O bloco destacado separa o sinal em componentes de **fase** e **quadratura**, que são então multiplicadas por seno e cosseno antes de serem processadas pelo subtrator. Seguindo a fórmula:

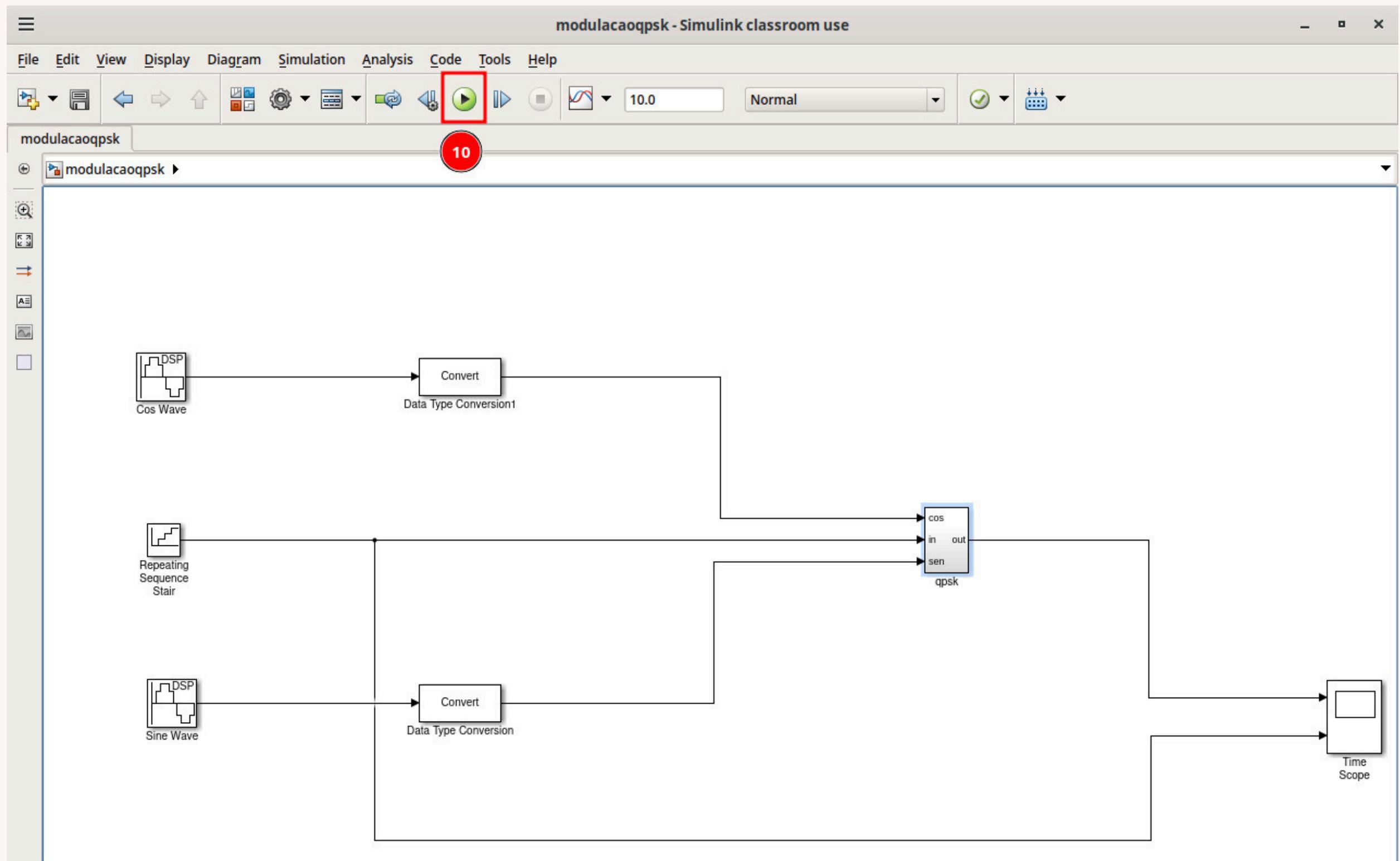
$$s(t) = I(t) \cdot \cos(2\pi f_c t) - Q(t) \cdot \sin(2\pi f_c t)$$

Desenvolvimento



9 - Volte para
o topo...

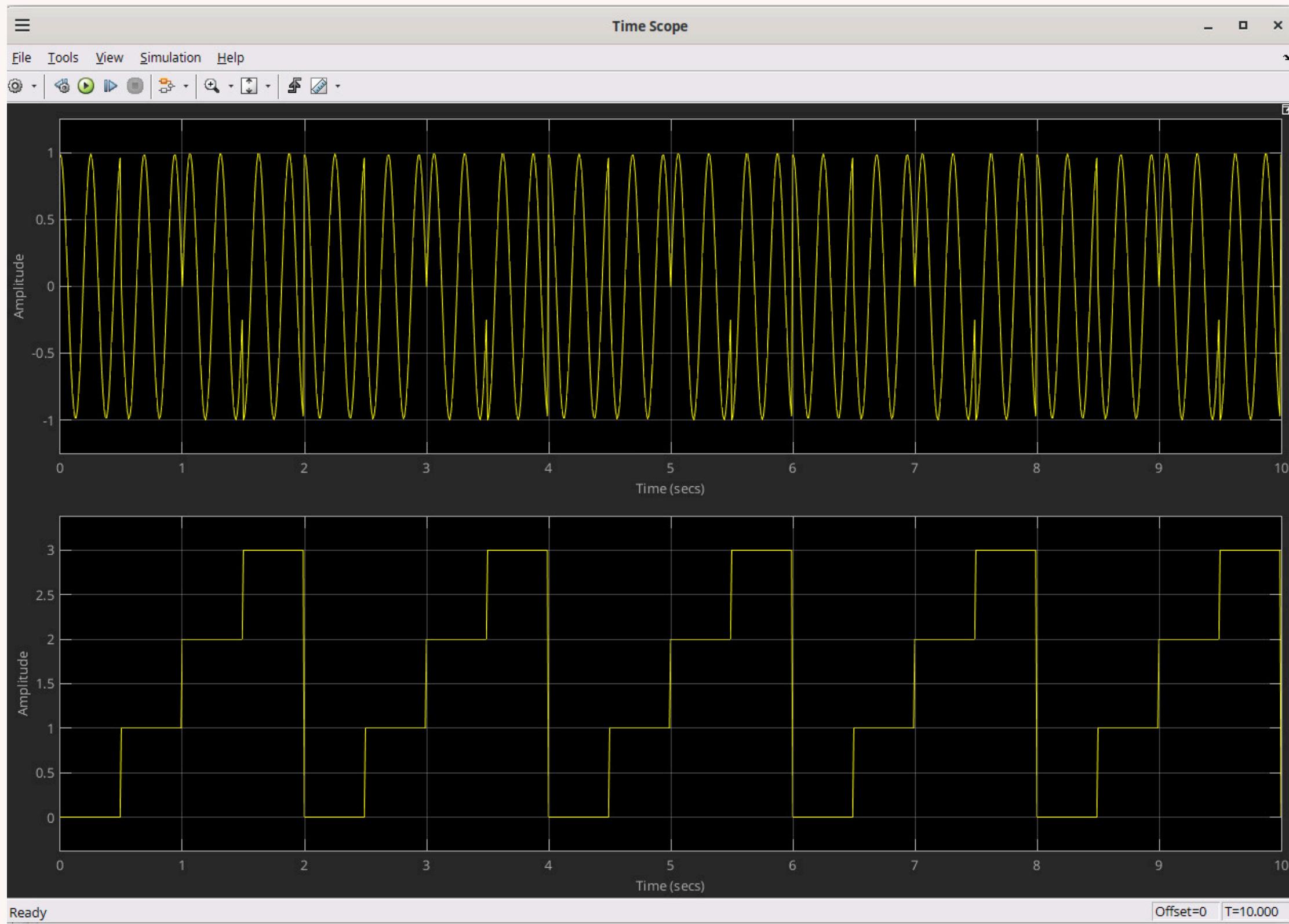
Desenvolvimento



10 - Clique em **run**



Desenvolvimento



Abrirá o Time Scope...

Note as mudanças de fase a cada mudança de nível (em cada $\odot 0,5$ seg.)

Desenvolvimento

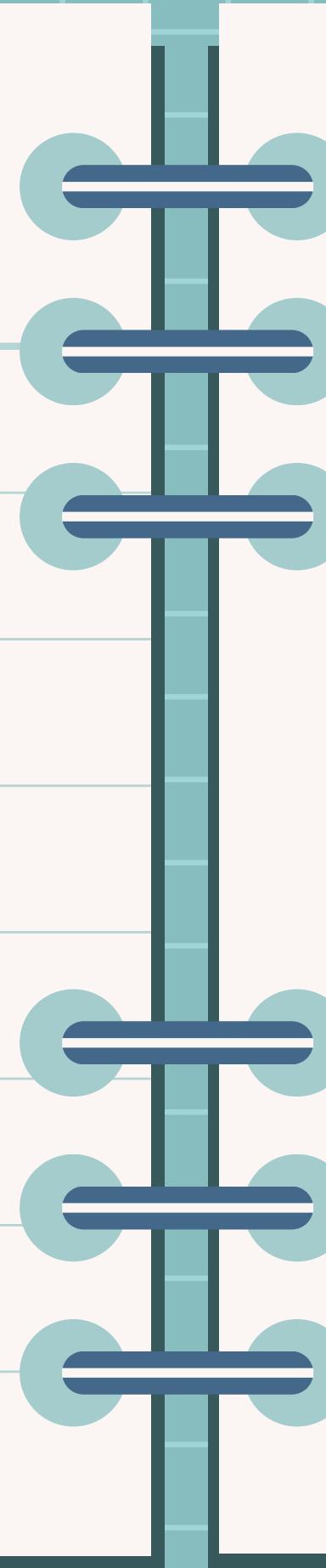
Pronto! A simulação agora implementa a **Modulação QPSK**. Neste esquema, a entrada digital é agrupada em símbolos (**valores 0, 1, 2 ou 3**), onde cada símbolo transporta dois bits de informação. O modulador utiliza duas portadoras ortogonais (Seno e Cosseno) para transmitir esses símbolos, resultando em uma das quatro possíveis mudanças de fase do sinal de saída: 0° , 90° , 180° ou 270° .



Atividades

Alterar a frequência da portadora

Objetivo: Avaliar como a variação da frequência da portadora altera a forma de onda da modulação QPSK, observando mudanças na densidade dos ciclos e no comportamento das transições de fase.



Tarefa:

- Simule a QPSK usando 4 Hz, depois 6 Hz e depois 8 Hz como frequência da portadora;
- Compare o sinal modulado.

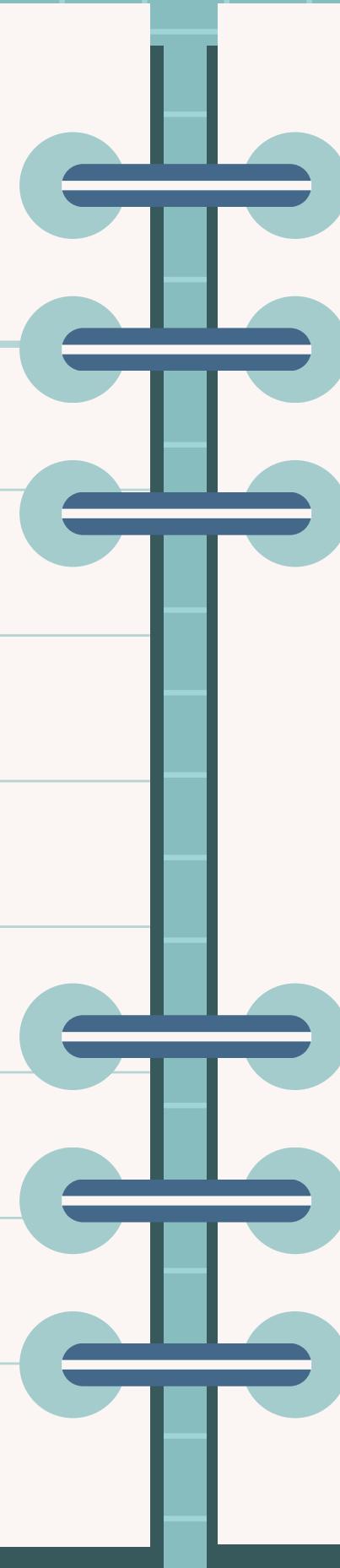
Pergunta:

O que muda? A forma de onda fica mais comprimida?

Atividades

Mudar o phase offset do modulador

Objetivo: Observar como a alteração do phase offset afeta a posição dos símbolos na constelação da QPSK.

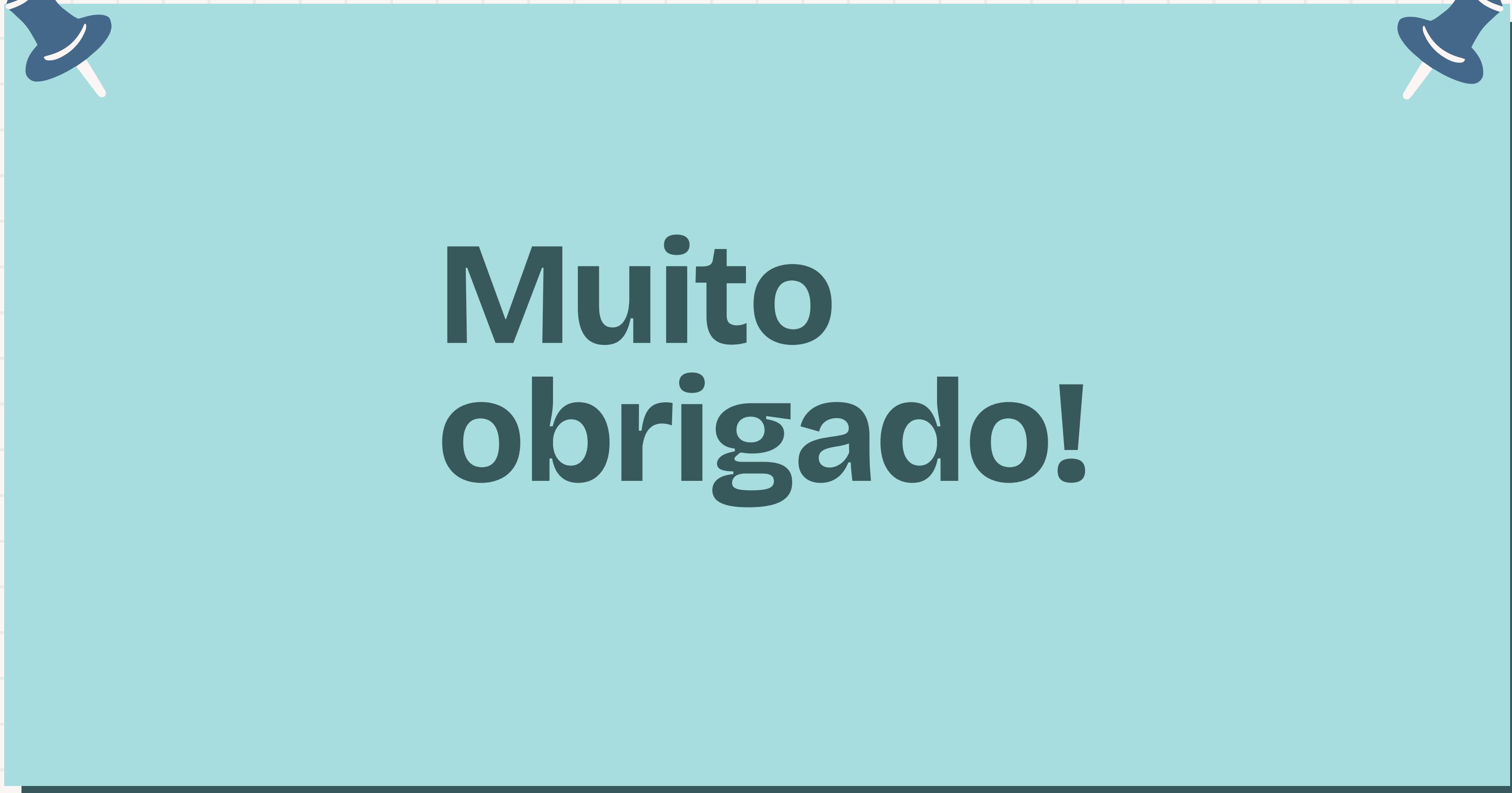
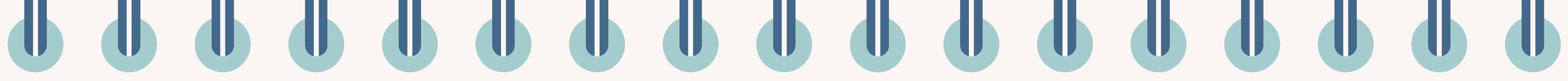


Tarefa:

- Simule a QPSK usando 4 Hz, depois 6 Hz e depois 8 Hz como frequência da portadora;
- Compare o sinal modulado.

Pergunta:

O que muda? A forma de onda fica mais comprimida?



**Muito
obrigado!**