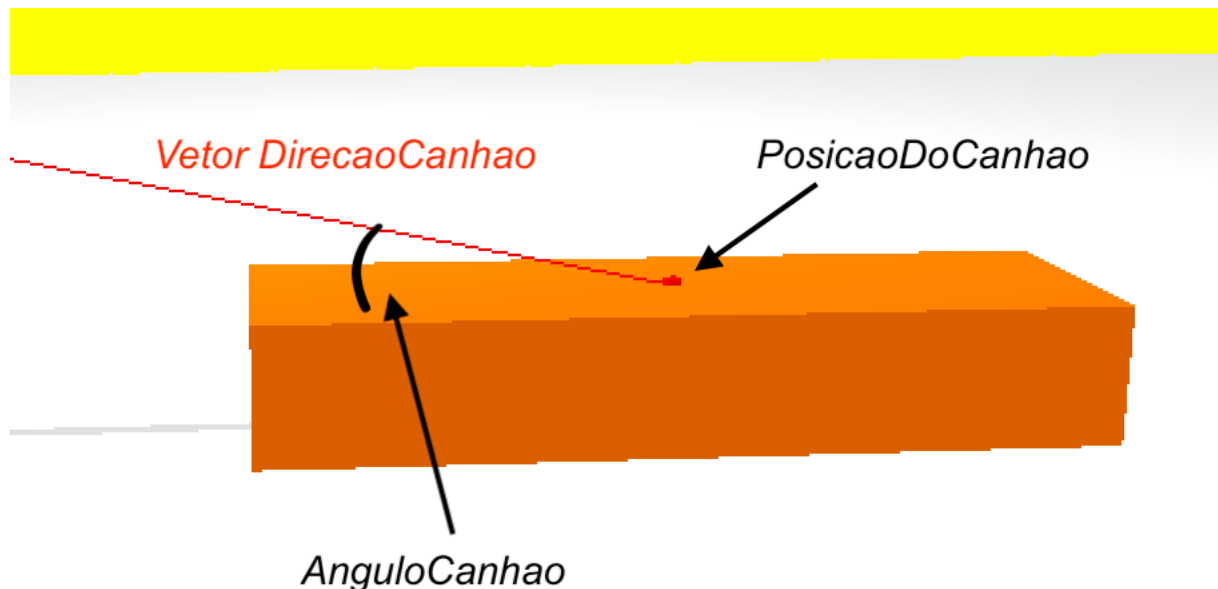


Pontos da trajetória do Projétil

Dados Iniciais

Para a formulação apresentada a seguir, assume-se que são mantidas atualizadas as seguintes informações:

- **PosicaoVeiculo**: posição do veículo
 - Deve ser atualizada da mesma forma que foi feito com o polígono no Trabalho 1.
- **AnguloVeiculo**: ângulo de rotação do veículo ao redor do eixo Y
- **AnguloCanhao**: ângulo de rotação do canhão ao redor do eixo Z

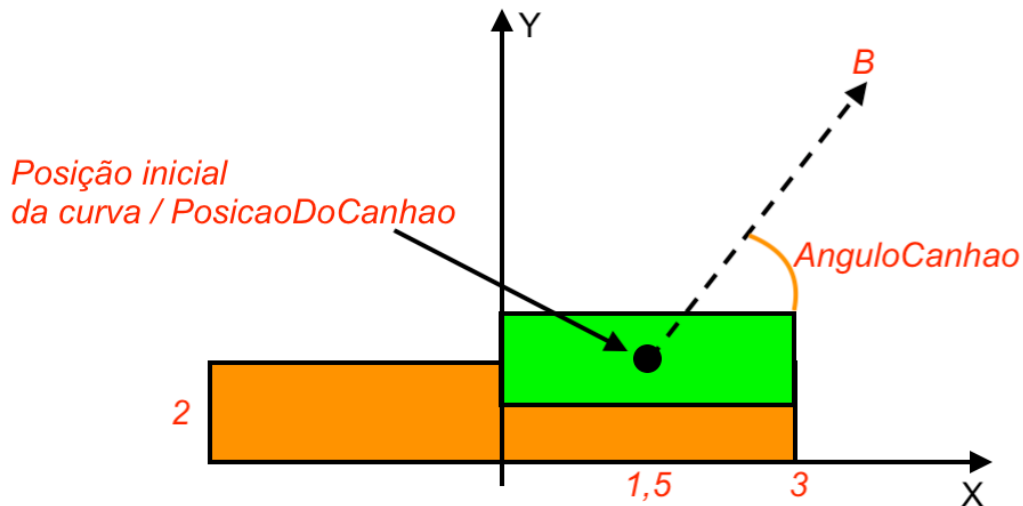


- **DirecaoDoCanhao**: vetor direção do canhão
 - Deve iniciar como Ponto (1,0,0). Sempre que um dos ângulos **AnguloVeiculo** ou **AnguloCanhao** forem alterados, deve-se atualizar a **DirecaoDoCanhao**, da seguinte forma.

```
Ponto DirecaoDoCanhao = Ponto(1,0,0);  
DirecaoDoCanhao.rotacionaZ(AnguloCanhao);  
DirecaoDoCanhao.rotacionaY(AnguloVeiculo);
```

Ponto Inicial da Trajetória

Calcule a posição do início da trajetória aplicando translações na **PosicaoVeiculo**. Na figura a seguir, por exemplo, o deslocamento seria (1.5, 2.0, 0.0)



Ponto **PosicaoDoCanhao**;

PosicaoDoCanhao = **PosicaoVeiculo** + Ponto (1.5, 2.0, 0);

Para desenhar uma linha que represente o primeiro segmento da trajetória, pode-se fazer uma linha entre **PosicaoDoCanhao** e B, sendo

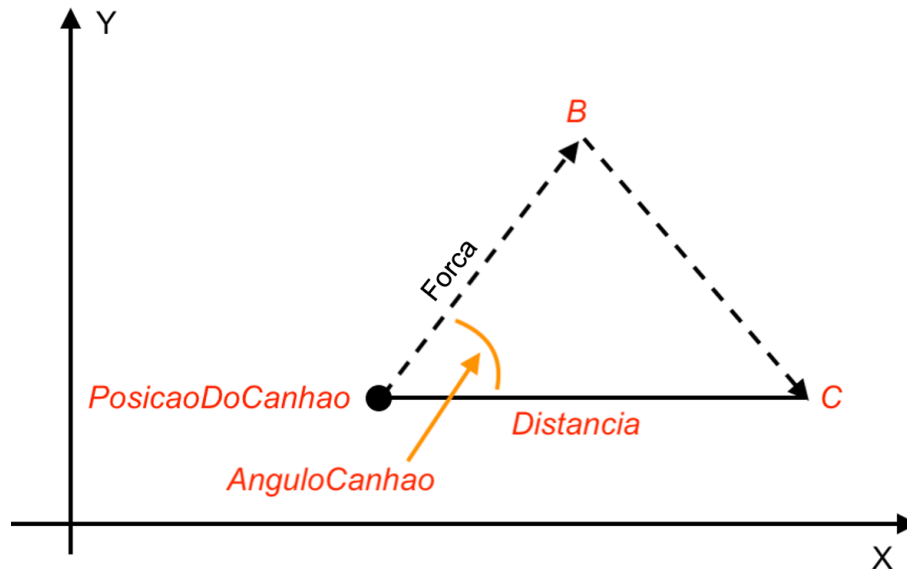
B = **PosicaoDoCanhao** + **DirecaoDoCanhao** * **forca**;

Neste exemplo, a variável **forca** define a distância entre **PosicaoDoCanhao** e B e, consequentemente, o alcance da trajetória do projétil.

Ponto Final da Trajetória

Considerando uma vista frontal, o ponto C, da figura a seguir, pode ser calculado por

```
Ponto C = PosicaoDoCanhao + Ponto (Distancia, 0,0);  
C.rotacionaY(AnguloVeiculo); // para alinhar com o veículo
```



Considerando que este é um triângulo isósceles (2 lados iguais), a distância entre os pontos **PosicaoDoCanhao** e **C** é dada por:

$$\text{Distancia} = 2 * \text{forca} * \cos(\text{AnguloCanhao} * 3.14 / 180);$$

Em 3D, o cenário fica desta forma:

