Relatório sobre o detalhamento do algoritmo e código usado

Altura do quadrirrotor

```
/-------------------------------------------------------------------------\
Dados
Distância > Alcance(3,0 metros)  --------------- velocidade linear z = - 1,0 m/s
Distância > Altura máxima(2,2 metros) ––––––––– velocidade linear z = - 1,0 m/s
Distância > Altura limite superior(1,8 metros)-- velocidade linear z = - 0,6 m/s
Distância < Altura limite inferior(1,6 metros)-- velocidade linear z =   0,6 m/s
\-------------------------------------------------------------------------/
```

Para definir a altura ideal que o quadrirrotor deveria ficar, eu comecei olhando os limites que o quadrirrotor pode chegar, que é 3 metros, alcance do sensor. Essa altura é praticamente a mesma dos corredores do departamento de elétrica, entretanto, no meio dos corredores existe um suporte para as luzes que fica 2,2 metros distante do chão. Logo, eu defini que a altura máxima do quadrirrotor deveria ser 1,8 metros, assim existe uma grande margem entre o robô e as luzes. Para a altura mínima, eu coloquei 1,6 metros, assim o robô consegue ficar na mesma faixa da cabeça de uma pessoa comum.

Obs.: quando eu diminuía a distância entre a altura máxima e mínima, o robô oscilava muito e não permanecia na faixa de altura desejada (eu observei essa oscilação com uma faixa de distância de 10 cm), mas quando eu aumentei essa distância para 20cm, ou seja, altura máxima 1,8 e mínima 1,6 metros, essa oscilação praticamente desapareceu.

Para receber os dados dos sensores ultrassônicos, eu inscrevi o nodo "quadrotor_node" criado no tópico /sonar_height e peguei essa informação na instância "range" da estrutura do tópico. Com essa informação eu pude saber a distância do quadrirrotor para o chão em metros, assim eu coloquei para o robô subir até a distância desejada que seria entre 1,8 metros e 1,6 metros. Para fazer o robô subir eu apenas coloquei 0,6 m/s no vetor de velocidade de linear no eixo Z. Agora, caso o robô ultrapassasse a altura de 1,6 metros o robô não iria mais receber essa velocidade e, se o robô ultrapasse o limite superior de 1,8 metros, o robô receberia a mesma velocidade no mesmo vetor, na mesma direção, porém sentido contrário, -0,6 m/s. Um detalhe é que se o robô ultrapassasse a altura das lâmpadas de 2,2 metros ou o alcance máximo do sensor de 3,0 metros essa velocidade aumentaria para -1,0 m/s. Esses valores de velocidade foram publicados no tópico /cmd_vel.

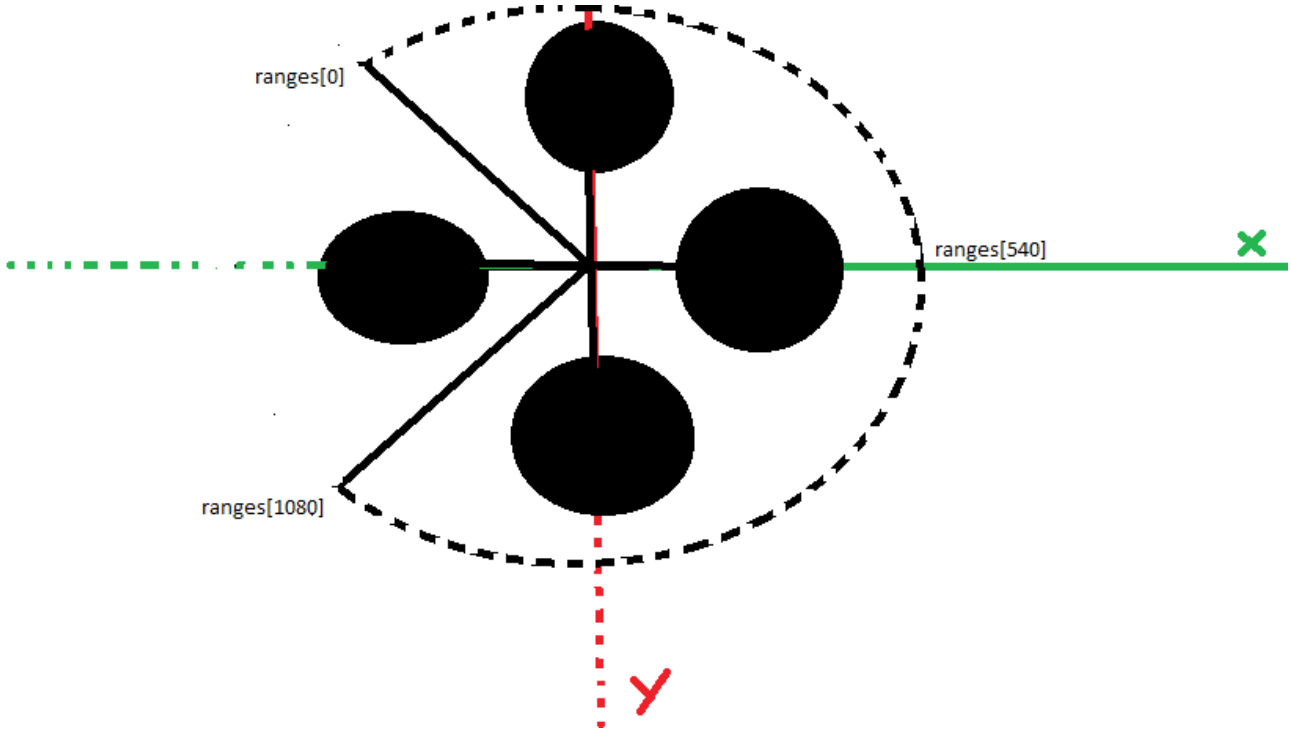
Detecção das paredes e objetos nas laterais do robô

```
/-------------------------------------------------------------------------\
Dados
Distância < limite lateral esquerdo(0,55 metros)-- velocidade linear y =-0,2 m/s
Distância < limite lateral direito(0,55 metros)-- velocidade linear y = 0,2 m/s
\-------------------------------------------------------------------------/
```

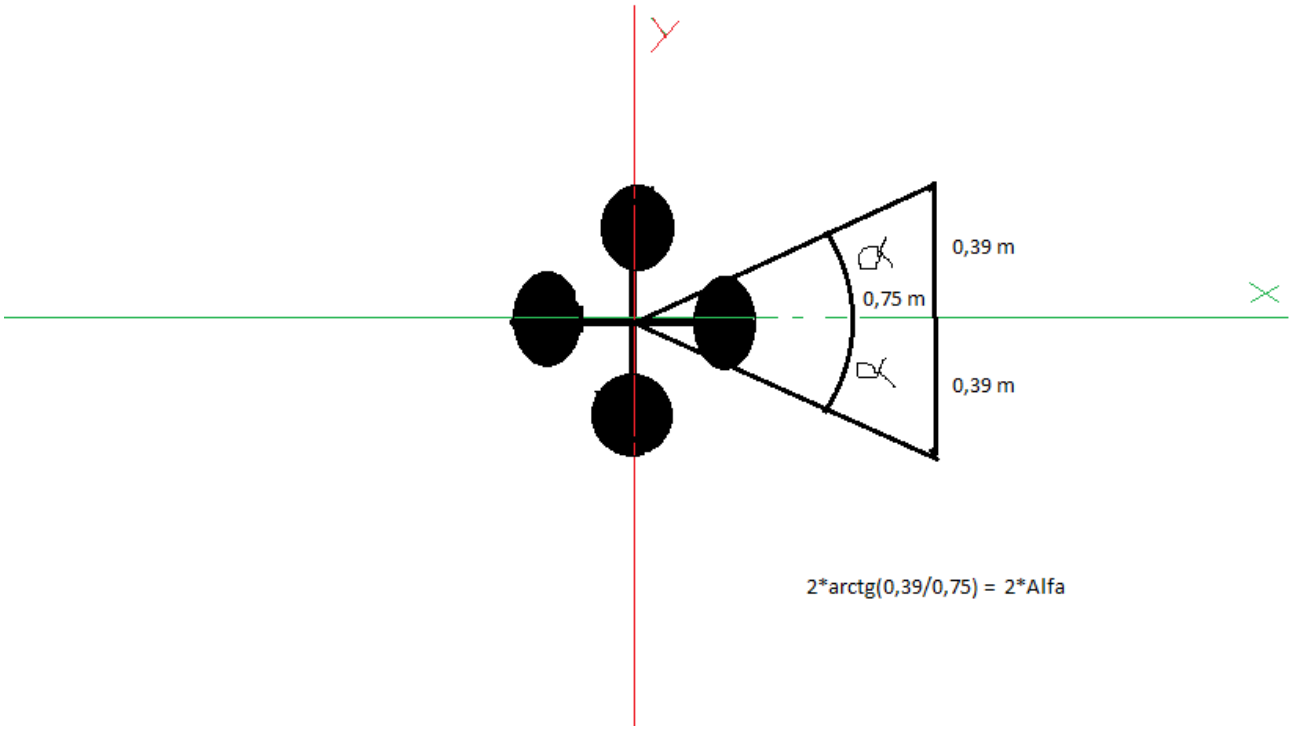
Para detectar as paredes e objetos que podem aparecer nas laterais do quadrirrotor eu utilizei os dados do tópico /scan. Esse tópico mostra o alcance do sensor que neste caso é de 270º, divididos pelo eixo x da parte frontal do robô, assim o sensor alcança 135º para cada lado do robô partindo do eixo x. A mensagem possui uma instância chamada “ranges” que é um vetor de 1080 pontos flutuantes com as dados de cada ângulo medido pelo sensor.

A explicação da mensagem do tópico que eu tinha visto dizia que o dado “ranges[0]” começava pela parte diagonal traseira direita do robô, mas depois de testar o quadrirrotor eu percebi que era o contrário, “ranges[0]” fica na parte diagonal traseira esquerda do robô. O desenho abaixo mostra melhor como funciona

esse vetor “ranges[0]”.



Tendo conhecimento disto, eu medi a largura do robô que é de 0,78 m , aproximadamente, e calculei qual seria a angulação necessária para o robô ficar a uma certa distância da parede, 0,75 m. Eu encontrei que seria necessário de um ângulo de 56º para este objetivo. Assim, para o lado direito o nodo receberia os dados do ângulo de 17º até 73º, já para o lado esquerdo o nodo receberia os dados do ângulo de 197º até 253º. Com essas informações, eu pude colocar uma velocidade linear y de 0,2 m/s, caso a parede direita estivesse perto demais do robô, menor que 0,75 m , e uma velocidade linear y de -0,2 m/s, caso a parede esquerda estivesse perto demais do robô, menor que 0,75 m. Entretanto, eu notei que o robô oscilava bastante, pois o robô conseguia apenas ficar a 0,6 m distante da parede, no máximo, assim eu diminui esse limite de distância para 0,55 m. No teste, o robô apresentou uma maior estabilidade que pode ser vista no vídeo https://youtu.be/AbiNLHjYTPM.



Detecção de paredes e objetos na frente do robô assim como sua locomoção no eixo x

```
/-------------------------------------------------------------------------------\
Dados
Distância > (0,75 metros) ---------- velocidade linear x = 0,5 m/s
Distância < limite frontal desejado(0,75 metros)-- velocidade linear x = 0,0 m/s
Distância < limite frontal extremo(0,4 metros)-- velocidade linear x =  -0,3 m/s
\-------------------------------------------------------------------------------/
```

Para a detecção de obstáculos a frente do robô, eu utilizei a mesma técnica usada para a detecção das paredes e obstáculos laterais, ou seja, os dados processados do sensor laser foram em uma faixa de 56º partindo do ângulo 107º até 163º, sendo que o quadrirrotor deveria manter uma distância entre 0,75 m e 0,4 m, caso o quadrirrotor ultrapassasse a distância de 0,75 m a velocidade linear x do robô seria nula, caso acontecesse de o quadrirrotor chegar perto o suficiente de um objeto para que a distância entre os dois fosse menor que 0,4 m o robô receberia uma velocidade linear x de -0,3 m/s, mas se essa distância fosse maior que 0,75 m o robô receberia uma velocidade x de 0,5 m/s.

Nos testes o robô inclinou para frente e para trás principalmente no momento de parada, no resto da simulação o quadrirrotor se comportou de maneira estável. Eu também testei com velocidades mais altas como de 3,0 m/s e eu pude notar que o quadrirrotor apesar de conseguir desviar das paredes laterais com sucesso, ele chegava muito próximo a elas devido ao tempo de resposta (as velocidades y continuaram sendo +-0,2 m/s). Outro fator foi no momento de chegava do quadrirrotor que não teve tempo suficiente para parar e colidiu com a parede.

Detalhes do quadrirrotor e ambiente de simulação usado

O modelo de quadrirrotor utilizado foi o disponível no pacote hector, mais especificamente, o incluído na demo do quadrotor no ambiente interno.
Os dados dessa demo podem ser encontrados no arquivo indoor_slam_gazebo.launch contida na pasta hector_gazebo_demo.

```
/-------------------------------------------------------------------------\
include file="$(find hector_quadrotor_gazebo)/launch/spawn_quadrotor.launch" >
    <arg name="model" value="$(find
hector_quadrotor_description)/urdf/quadrotor_hokuyo_utm30lx.gazebo.xacro"/>
  </include>
\-------------------------------------------------------------------------/
```

Foi utilizando o ambiente do escritório da willow garage, contida nessa demo, que foram feitos os testes com o quadrirrotor. Não foi utilizada os sistemas de SLAM nem o Geotiff mapper.