

Sistema FIEB



PELO FUTURO DA INOVAÇÃO

CENTRO UNIVERSITÁRIO SENAI CIMATEC

Relatório Final

Desafios para o laboratorio robotica e sistemas autonomos

Apresentada por: Tiago Barretto Sant'Anna

Orientado por: Prof. Marco Reis, M.Eng.

Dezembro de 2021

Tiago Barretto Sant'Anna

Desafios para o laboratorio robotica e sistemas autonomos

Salvador
Centro Universitário SENAI CIMATEC
2020

Resumo

Escreva aqui o resumo da dissertação, incluindo os contextos geral e específico, dentro dos quais a pesquisa foi realizada, o objetivo da pesquisa, assunção filosófica, os métodos de pesquisa usados e as possíveis contribuições que o que é proposto pode trazer à sociedade.

Palavras-chave: Palavra-chave 1, Palavra-chave 2, Palavra-chave 3, Palavra-chave 4, Palavra-chave 5

Abstract

Escreva aqui, em inglês, o resumo da dissertação, incluindo os contextos geral e específico, dentro dos quais a pesquisa foi realizada, o objetivo da pesquisa, assunção filosófica, os métodos de pesquisa usados e as possíveis contribuições que o que é proposto pode trazer à sociedade.

Keywords: Keyword 1, Keyword 2, Keyword 3, Keyword 4, Keyword 5

Sumário

Lista de Figuras

Lista de Tabelas

Introdução

Este artigo consiste na exposição dos desafios realizados para o laboratório de Robotica e Sistemas Autonomos. Assim esses desafios foram realizados ao longo de dois meses. Esses desafios tem como principal função realizar a capacitação para pode atuar dentro do laboratório

1.1 *Objetivos*

Os objetivos são realizar desafios de robotica, programação e simulação com a função de adquirir conhecimentos para poder atuar dentro do laboratório.

1.1.1 *Objetivos Específicos*

Os objetivos específicos deste projeto são:

- Aprender ROS e como utiliza-lo;
- Desenvolver habilidades de programação em Python e C++;
- Obter conhecimento de simulação e programação de robôs usando o Webots;
- Criar um package no ROS para controlar o turtlesim;
- Obter conhecimentos a cerca de navegação usando o Husky

1.2 *Justificativa*

Esse trabalho tem como função de capacitar o estudante a cerca das ferramentas necessarias para trabalhar com robotica, sendo aprendendo a utilizar o framework ROS com a versão noetic, ou habilitando seu conhecimento em programação.

1.3 Organização do documento

Este documento apresenta 5 capítulos e está estruturado da seguinte forma:

- **Capítulo ?? - Introdução:** Contextualiza o âmbito, no qual a pesquisa proposta está inserida. Apresenta, portanto, a definição do problema, objetivos e justificativas da pesquisa e como este relatório final está estruturado;
- **Capítulo ?? - Fundamentação Teórica:** Explicita toda a base teórica com o qual o trabalho foi produzido, tratando das bases para a sua construção;
- **Capítulo ?? - Materiais e Métodos:** Evidencia por qual processo foi executado o objetivo desse artigo até a obtenção dos resultados;
- **Capítulo ?? - Resultados:** Mostra o que foi obtido com a realização desse trabalho;
- **Capítulo ?? - Conclusão:** Apresenta as conclusões, contribuições e algumas sugestões de atividades de pesquisa a serem desenvolvidas no futuro.

Conceito do projeto

Para a realização de cada desafio foi necessário captar conhecimento de diversas fontes para poder construí-lo. Contudo fica evidente a necessidade de trazer esses materiais que inspiraram a produção de tal material.

2.1 *Desafio Workbooks Python*

O desafio de Workbooks utilizando a linguagem python foi composto de 16 desafios de programação utilizando a linguagem python. Partindo dessa contexto, tudo se iniciou com o estudo de bibliotecas e python, para poder realizar as tarefas da forma mais eficiente possível.

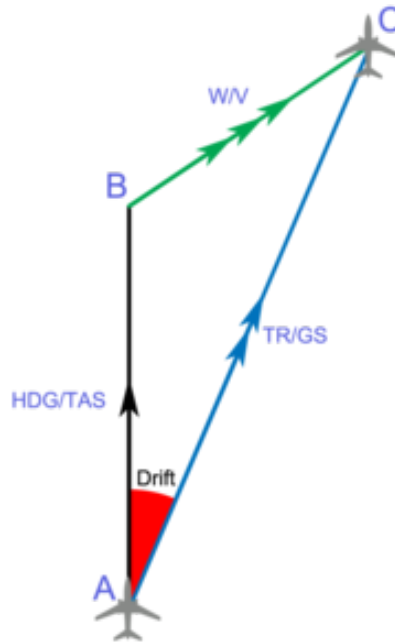
2.2 *Desafio C++*

O desafio de C++ foi composto de três desafios de programação que utiliza a linguagem de programação C++. Portanto para a primeira tarefa foi feito um estudo matemático para entender como funciona o triângulo de pascal. Em seguida para a segunda tarefa foi feito um estudo combinatório sobre permutações e depois um estudo sobre bibliotecas com a função de realizar essas combinações de palavras de forma mais eficaz. Por fim, para a terceira foi feito um estudo acerca de bibliotecas que trabalhassem com arquivos para assim ler o código que estava escrito. Dessa maneira, foram baseados a construção dos desafios.

2.3 *Turtlesim setpoint position*

Consiste em utilizar o software *Turtlesim* para através do ROS escolher uma posição e fazê-lo se deslocar até ela. A movimentação programada na *Turtlesim* se baseia em uma teoria principal, que é o *Dead Reckoning*. Dessa forma, o *Dead Reckoning* consiste em calcular a distância que falta do robô para determinado objetivo e incorporar esses valores a sua posição e velocidade, como pode ser exemplificado na figura ???. No caso da *Turtlesim* foi calculado sua distância usando o teorema de pitágoras e o ângulo pelo qual a tartaruga precisou virar pela trigonometria.

Figura 2.1: Dead reckoning



Fonte: Wikipedia.

Na navegação do *Turtlesim*, o teorema de pitagoras é utilizado para calcular a distância entre a mesma e o objetivo definido durante cada instante. Para isso foi utilizado a fórmula $d = \sqrt{(x_f - x_i)^2 + (y_f - y_i)^2}$. Nos quais x_f e y_f são as distâncias finais e x_i e y_i são as distâncias da tartaruga em determinado instante.

Assim, para calcular quanto que o *Turtlesim* precisa girar faz o uso da trigonometria, para calcular o ângulo do qual a *Turtlesim* esta deslocada em relação ao objetivo. Com isso, se faz o uso da fórmula de arctg na qual $\theta = \frac{y_f - y_i}{x_f - x_i}$. Por fim, pegamos esse angulo e subtraímos do ângulo atual da *Turtlesim* para descobrir o quanto a tartaruga precisa rotacionar.

Para definir a velocidade da tartaruga multiplicamos a distância por uma constante e para descobrir o quanto ela precisa rotacionar multiplicamos a angulação por outra constante. Assim esses dados são publicados no topic `cmd_vel` para alterar a velocidade da turtle até que ela chegue no objetivo especificado.

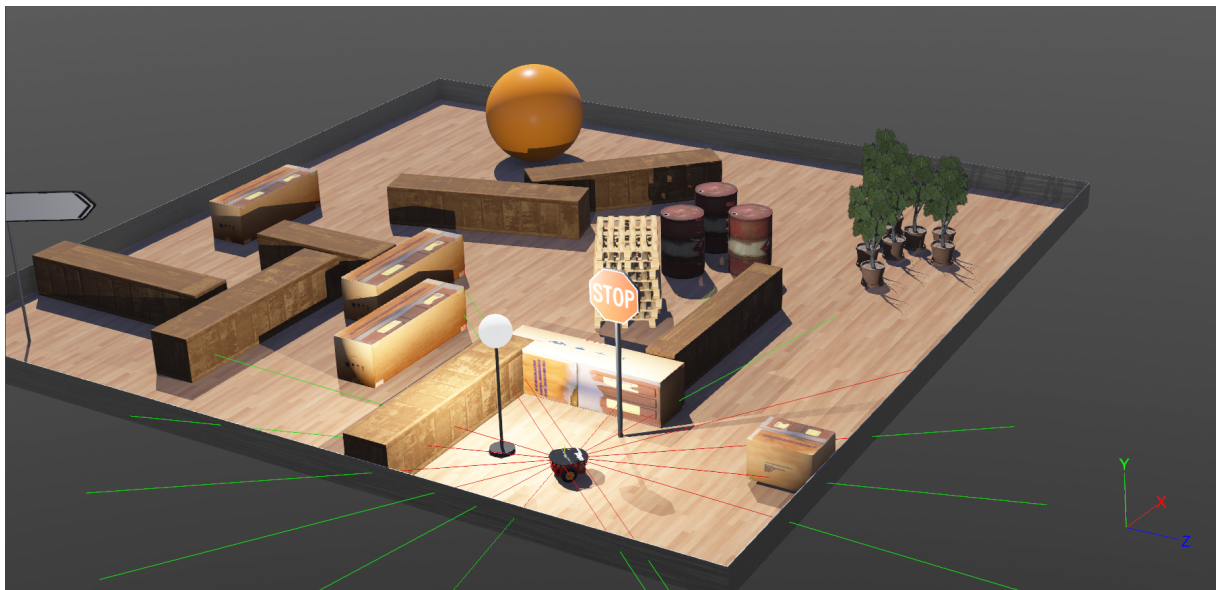
2.4 Desafio Webots

O Webots é uma plataforma opensource usada para simular robôs. Desse modo, o desafio consiste em utilizar essa plataforma para simular um robo chamado pioneer3x, corrigindo

o código já existente e alterando ele para caso ele encontre uma luminaria pare de se locomover.

A partir disso foram realizados os tutoriais dessa plataforma para ter conhecimento de como utilizá-la e como simular os robôs. Em seguida foi colocado um sensor de luz no pioneer3x para que o mesmo tenha uma forma de detectar a luminosidade do ambiente e seu código foi alterado para que se a leitura do sensor ultrapasse determinado valor ele pare.

Figura 2.2: Webots



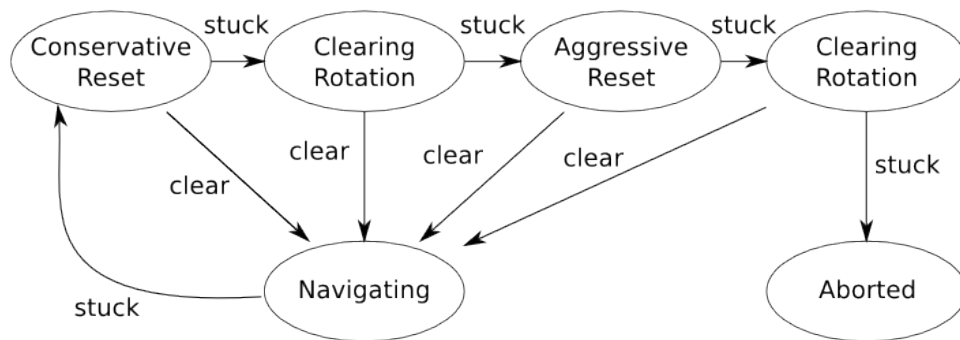
Fonte: Autoria propria.

2.5 *Desafio Husky*

Husky é um veículo UGV no qual através do ROS pode ser simulado em conjunto com o gazebo e o rviz, sendo ambos simuladores no qual o primeiro é voltado para o ambiente ao redor do husky e o segundo é como esse ugv percebe o mundo. O desafio consiste em utilizar os simuladores para testar diferentes formas de navegação com o husky.

O primeiro desafio é simular o husky com o package move base. Consiste em dar uma localização no mundo e ele irá tentar atingir esse objetivo. Caso o ugv identifique algum obstáculo ele irá desviar, ou caso fique preso irá entrar em um processo chamado conservative reset se parar de ficar preso voltará a navegação, caso não entrara em clearing rotation, se mesmo assim continuar preso iniciará um aggressive reset e continuando preso vai por fim fazer uma clearing rotation e se mesmo assim continuar preso vai abortar a ação. Como mostra na imagem ??.

Figura 2.3: Move Base

move_base Default Recovery Behaviors

Fonte: ROS Wiki.

O segundo desafio é o amcl demo, o qual é a junção do move base com o amcl. Assim o amcl é um sistema probabilístico de localização do robô o qual através de sensores de laser fazem o tracking da posição do robo dentro de um mapa.

Gmapping demo é o terceiro desafio, ele é a junção do move base com o gmapping. Esse package prove um SLAM (Simultaneous localization and mapping), baseado em sensores a laser. Com o gmapping é criado um mapa 2D do ambiente. Como pode ser mostrado na imagem abaixo.

Por ultimo foi realizado o frontier exploration demo, é composto pelo move base, gmapping, e o frontier exploration. Dessa forma, o frontier em conjunto com esses packages para realizar a exploração de ambientes

Desenvolvimento do projeto

Durante esta seção será descrito o processo de construção dos desafios, incluindo os estudos necessários para cada um, o processo de criação e especificidades. Será apresentado estas características para cada um desafio.

3.1 Desafio Workbooks Python

Para realização deste desafio foi realizado a pesquisa sobre bibliotecas em python para cada tarefa propria e em seguida criação de scripts e testes. Para dessa forma concluir o desafio.

3.2 Desafio C++

Tem como inicio o estudo da linguagem C++ através da Code Academy com o seu curso. Depois, para cada tarefa foi realizado uma pesquisa propria, sendo sobre matematica teoria ou bibliotecas dessa linguagem. Assim, foi realizado o desafio.

3.3 Turtlesim setpoint position

O desafio se iniciou com um estudo sobre o ROS, mais especificamente sobre publisher e subscriber. Após isso, foi feito uma pesquisa acerca das tecnicas de navegação chegando no dead reckoning. Em seguida, iniciou-se a fase de estudo da programação e testes do package ate chegar no produto final.

3.4 Desafio Webots

Em primeira etapa foram realizados os tutoriais do Webots, logo em diante foi feito um estudo da programação nesta plataforma. Depois disso foram realizados seguidos testes e alterações ate que o robo conseguisse se locomover corretamente para apos isso implementar o sensor de luz.

3.5 *Desafio Husky*

O desafio foi realizado com o uso do ROS Noetic. Para ser realizado foi feito *git clone* de um repositório no github do Husky em um workspace. Com isso, foi realizados os desafios porém alguns erros surgiram, mas foram solucionados através de pesquisa e ajuda de colegas.

Resultados

Depois de finalizados os desafios foram coletados os seus resultados e disponibilizados no github.

4.1 Desafio Workbooks Python

Com a conclusão do desafio os códigos foram testados e tiveram sucesso no seu funcionamento. Assim,

4.2 Desafio C++

Os códigos foram compilados e executados obtendo sucesso no seu funcionamento. O primeiro

4.3 Turtlesim setpoint position

Depois de finalizado o package tinha seu funcionamento como exigia o regulamento. Portanto as coordenadas eram digitadas e a tartaruga realizava um deslocamento até alcançar uma distância de pelo menos 0.1 do objetivo e assim encerrava seu programa.

4.4 Desafio Webots

Após o explicitado na metodologia o robô Pioneer3X conseguiu seguir o percurso sem grandes problemas. Desse modo e parar quando encontrava uma fonte luminosa, sendo no caso uma luminária em um canto do mapa dentro de XX segundos

4.5 *Desafio Husky*

As diferentes navegações foram corretamente executadas e colocadas em prática dentro dos simuladores gazebo e rviz. Com isso foi capaz perceber como as diferentes técnicas de navegações aplicadas e combinadas afetam o deslocamento do ugv

Conclusão

Durante todos os desafios foram exigidos o aprendizado e dominio de novas ferramentas. Os desafios foram compostos de uma serie de desafios envolvendo programação, simulação, navegação e dominio do *ROS*.

5.1 Considerações finais

O trabalho fomentou um aprendizado especifico de maneira pratica e objetiva, voltado para a criação de habilidades, o desenvolvimento do pensamento logico para resolução de problemas e absorção das ferramentas utilizadas

Sendo assim, essas ferramentas são de grande importância para o trabalho e desenvolvimento em robótica

Diagramas mecânicos

Diagramas eletro-eletrônicos

Desafios para o laboratorio robotica e sistemas autonomos

Tiago Barretto Sant'Anna

Salvador, Dezembro de 2021.