

Mapeamento 2D Por Sensoreamento à Laser

Otávio Gonçalves Vicente Ribeiro Filho

Submetido ao Departamento de Engenharia Elétrica
como trabalho final de graduação do curso de

Engenharia de Computação

UNIVERSIDADE FEDERAL DA BAHIA

Janeiro 2017

Autor.....
Departamento de Engenharia Elétrica
21 de Dezembro de 2016

Orientador.....
André Scolari
Associate Professor

Banca examinadora.....
Simas
Paulo Cesar
Wagner

Mapeamento 2D Por Sensoreamento à Laser

Otávio Gonçalves Vicente Ribeiro Filho

Submetido ao Departamento de Engenharia Elétrica
em 21 de Dezembro de 2016, como trabalho final de graduação em
Engenharia de Computação

Resumo

Neste trabalho será apresentado um mapeamento automatizado de um ambiente por um robô, utilizando um sensor de distância à laser, resultando por fim em mapa métrico em duas dimensões do ambiente mapeado.

Para tanto foi utilizando o V-Rep como ferramenta de simulação robótica, o Matlab para execução do algoritmo descrito em [Karsten Berns]. e o banco de dados relacional Microsoft SQLServer Express 2014 para armazenamento dos mapas obtidos ao fim de cada simulação.

Orientador: André Scolari
Título: Associate Professor

Agradecimentos

Primeiramente agradeço à minha família pelo apoio incondicional durante todas as minhas vidas. Especialmente ao meu pai, Otávio, que fez despertar em mim o interesse pela engenharia, a sempre perseguir meus objetivos e nunca desistir frente aos obstáculos no meio do caminho. Agradeço também à minha mãe, Ednalva, pelo carinho, atenção e cuidado, sem os quais a jornada até aqui teria sido exponencialmente mais difícil. Por fim agradeço à minha namorada, por estar sempre ao meu lado, me ajudando e apoiando.

Sumário

1	Introdução	6
2	Fundamentação Teórica	7
2.1	Sensoreamento Laser	7
2.2	Localização mundial	7
3	Desenvolvimento	8
3.1	Software de Simulação Robótica V-Rep	8
3.1.1	Habilitação da API Remota	9
3.1.2	Conexão dos Elementos no V-Rep	11
4	Resultados	13
5	Conclusão	14
A	Códigos Fonte	15
B	Referências Bibliográficas	16

Capítulo 1

Introdução

Texto como descrito em

Capítulo 2

Fundamentação Teórica

Texto sobre fundamentação teórica

2.1 Sensoreamento Laser

Breve texto sobre sensoriamiento à laser

2.2 Localização mundial

Capítulo 3

Desenvolvimento

Este capítulo irá tratar da implementação prática deste trabalho. Para tanto, será dividido em 4 seções: o software de simulação de robótica V-Rep, o sensor à laser de distância Hokuyo, o modelo de robô utilizado Pioneer 3-DX e, por fim, a implementação do algoritmo descrito em 2.X no Matlab.

3.1 Software de Simulação Robótica V-Rep

O V-Rep trata-se de um software de simulação robótica desenvolvido pela empresa Coppelia Robotics que, até a data de apresentação deste documento, encontra-se em sua terceira versão. Mesmo em sua instalação básica, este software já conta com a modelagem de diversos robôs móveis e fixos, bem como uma grande variedade de sensores disponíveis no mercado e utilizados em projetos reais de robótica. Não obstante, o software também oferece uma série de modelos para a simulação do ambiente de trabalho do robô, como paredes, esteiras rolantes, dentre outros. Caso o software não ofereça algum modelo específico, este também tem a funcionalidade de poder modelar qualquer outro robôs que não esteja já incluído em sua biblioteca, e também de interfaces para operação dos robôs.

Os modelos implementados podem ser programados de 2 formas: através de scripts escritos em linguagem Lua no próprio V-Rep, através da tela de scripts. Ou através

da API remota, que como descrito em [1], tem suporte às seguintes linguagens de programação:

- C\C++
- Python
- Java
- Matlab
- Octave
- Urbi
- Lua

Neste trabalho foi utilizado o Matlab para fazer a programação dos modelos, e na seção seguinte será descrito como configurar o V-Rep e o Matlab para trabalhar em conjunto.

3.1.1 Habilitação da API Remota

A habilitação da API remota no lado do servidor, V-Rep, pode ser feita de forma global ou de forma individual, nos scripts do projeto sendo executado. Quando habilitada de forma global, a API pergunte controlar qualquer projeto aberto assim que o V-Rep é executado. Quando a habilitação é feita de forma individual, só é possível controlar o projeto que teve sua simulação iniciada. Sendo assim, com a habilitação individual não é possível utilizar comandos como `simxStartSimulation`

Para habilitar a API de forma individual basta abrir a tela de scripts, acessível através do menu *Tools*→*Scripts*, e adicionar no topo do script em que se deseja habilitar a API a seguinte linha, onde *<porta>* corresponde à porta da conexão:

```
simExtRemoteApiStart(<porta>)
```

Para habilitar a API de forma global primeiramente é necessário criar um arquivo chamado *remoteApiConnections.txt*, dentro da pasta raiz do V-Rep. Dentro deste arquivo devem estar contidas as seguintes linhas:

```
portIndex@_port = <porta>
portIndex@_debug = true/false
portIndex@_syncSimTrigger = true/false
```

Neste caso @ corresponde ao índice, que servirá para identificar os parâmetros introduzidos. O primeiro comando especifica a porta para aquele índice, o segundo habilita ou desabilita debug através da porta, e o último habilita ou desabilita execução síncrona do V-Rep com o programa externo através da porta especificada pelo índice.

Após a habilitação da API no lado do servidor, é necessário também habilitá-la junto ao programa que será utilizado para controlar os modelos do V-Rep. No caso deste trabalho, será explicado como configurar a API no Matlab. Primeiramente é necessário adicionar os seguintes arquivos na mesma pasta do projeto do Matlab:

- *remApi.m*
- *remoteApiProto.m*
- *remoteApi.dll*

Os arquivos *.m* podem ser achados na pasta <raiz_V-Rep>\V-REP_PRO_EDU\programming\remoteApiBindings\matlab\matlab. O arquivo *.dll* pode ser encontrado na pasta <raiz_V-Rep>\V-REP_PRO_EDU\programming\remoteApiBindings\lib\lib\64Bit ou 32Bit). É importante ressaltar que as bibliotecas com a extensão *.dll* são usada no sistema Windows, para o caso do Linux e do Mac OsX as extensões dos arquivos serão diferentes, porém estarão contidas na mesma pasta.

Após estes passos, o ambiente de trabalho já está configurado e é possível iniciar uma conexão com o V-Rep. Para tanto primeiramente é necessário criar um objeto de controle do V-Rep utilizando-se a função *remApi.m* importada previamente, e especificando a biblioteca também importada previamente. Após isto deve-se utilizar o objeto para iniciar a conexão com o V-Rep utilizando-se o comando `simxStart`, que retornará o id da conexão estabelecida ou -1 caso a tentativa de conexão tenha fracassado:

```
vrep = remApi('remoteApi')
id = vrep.simxStart('127.0.0.1', 19997, true, true, 2000, 5);
```

3.1.2 Conexão dos Elementos no V-Rep

Após a configuração do ambiente de trabalho, é necessário inserir modelos necessários ao projeto para que a simulação possa ser executada. O V-Rep possui um sistema de hierarquia do tipo *pai-filho*, sendo assim o objeto pai consegue obter e alterar propriedades do objeto filho, mas o objeto filho não consegue fazer o mesmo com o pai. No caso deste projeto foi utilizado o sensor de distância à laser Hokuyuo URG-04LX-UG01 acoplado ao robô Pioneer P3DX, para isso é necessário que o sensor seja filho do objeto pai robô. Caso esta hierarquia não seja obedecida, quando a simulação for iniciada, o robô se moverá mas o sensor não, mesmo que este tenha sido corretamente posicionado no local desejado. Para tornar um objeto filho de outro, é necessário posicioná-lo dentro da árvore hierárquica do objeto pai, como mostrado na figura 3-1.

Na figura 3-1 também é possível observar a posicionamento do sensor, que terá seu *frame* movimentado na mesma direção, sentido e velocidade da movimentação do *frame* do robô, o objeto pai.

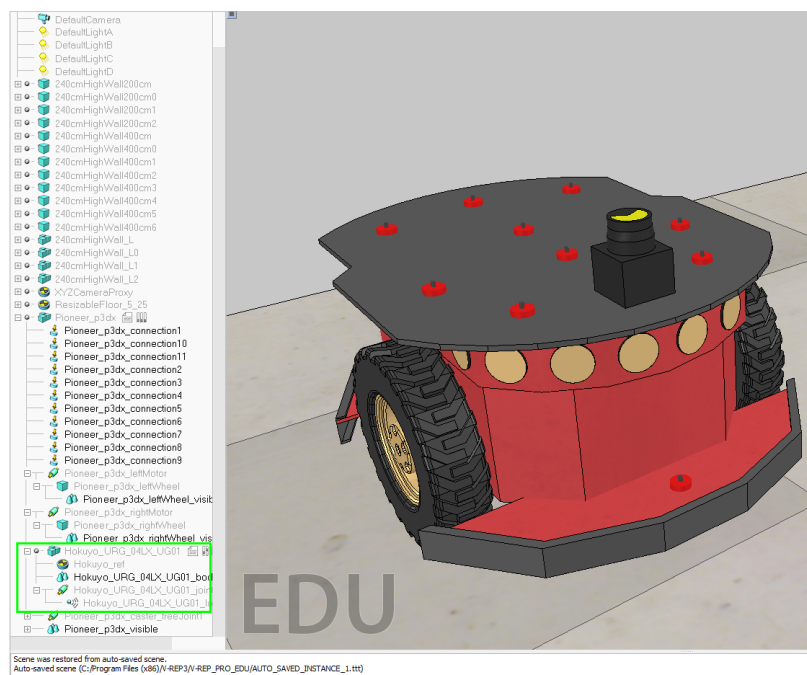


Figura 3-1: Árvore Hierárquica V-Rep

Capítulo 4

Resultados

Texto sobre os resultados obtidos, tamanho e organização das informações

Capítulo 5

Conclusão

Considerações finais sobre o trabalho

Apêndice A

Códigos Fonte

Tabela A.1: Armadillos

Armadillos	are
our	friends

Apêndice B

Referências Bibliográficas

- [1] Copelia Robotics. *Remote API*.