Universidade Federal do Rio Grande do Norte (UFRN)
Centro de Tecnologia
Departamento de Computação e Automação
Pablo Javier Alsina

Seminário 1

Dados de Identificação		
Comp. curricular:	DCA0440 -SISTEMAS ROBÓTICOS AUTÔNOMOS	
Data do exper.:	27/08/2019	
Componentes:	20211012651	Ednilson Pereira de Freitas
	20201025217	José Claercio Santos da Silva
	20201024980	Francisco Duarte Junior
	20201024882	William da Cunha Ribeiro

1 Resumo

O objetivo desse projeto é simular um robô móvel com acionamento diferencial e desenvolver um sistema de controle cinemático que permita ao mesmo executar movimentos especificados em espaço livre de obstáculos utilizando o Matlab.

2 Introdução

O V-REP(Virtual Robot Experimentation Platform) é uma plataforma cuja funcionalidade é simular robôs em ambientes integrados, podendo programar seus controladores em C/C++, Python, Java, Lua, MATLAB etc, portanto sendo bem versátil em sua utilização e ideal para múltiplas aplicações. O V-REP é utilizado para desenvolvimento rápido de algoritmos, simulação de automação de fábrica, prototipagem rápida e verificação, educação relacionada à robótica, monitoramento remoto, etc. Neste relatório será apresentado o desenvolvimento de um sistema de controle cinemático para simular robôs com acionamento diferencial.

3 Configuração da comunicação

3.1 Comunicação com o Matlab

Para realizar a comunicação com o matlab é bem simples, deve-se adicionar um pequeno comando no script do robô no Vrep e também no código do Matlab.

Non-threaded child script (RobotBase)

```
1  -- DO NOT WRITE CODE OUTSIDE OF THE if-then-end SECTIONS BELOW!! (unless the code is a function
2
3  if (sim_call_type==sim_childscriptcall_initialization) then
4      simExtRemoteApiStart(19999)
5
6      -- Put some initialization code here
7
```

Figura 1: Configuração no vrep

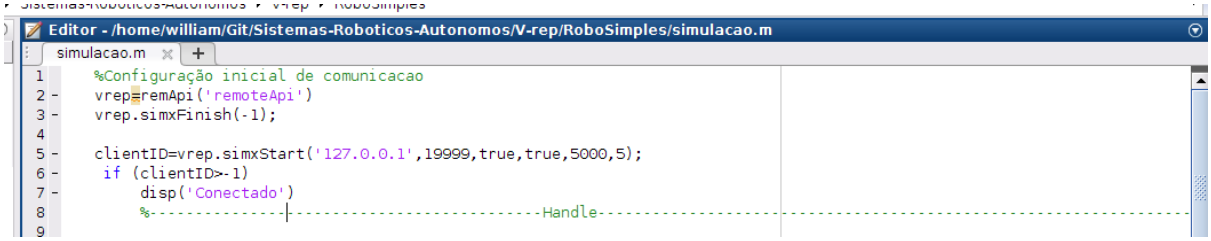

Figura 2: configuração matlab

Além disso, deverá acessar a pasta do V-Rep programming -> remoteApiBindings e você encontrará todas as linguagens que o V-Rep da suporte por meio do seu API. Como estamos usando o matlab, iremos entrar em sua pasta e copiar os arquivos dela para nossa pasta de trabalho, onde está a simulação do V-Rep. E por ultimo, precisaremos de mais um arquivo, que se encontra dento de remoteApiBindings, que já acessamos antes, só que dessa vez não iremos escolher o matlab, vamos em lib -> lib escolher o tipo do seu sistema, se é 32 ou 64 bits e copiaremos o arquivo que está dentro da pasta, nome do de remoteApi.so e também adicionaremos a pasta principal de nosso projeto. Feito isso, toda a configuração da comunicação do vrep com o matlab está finalizada.

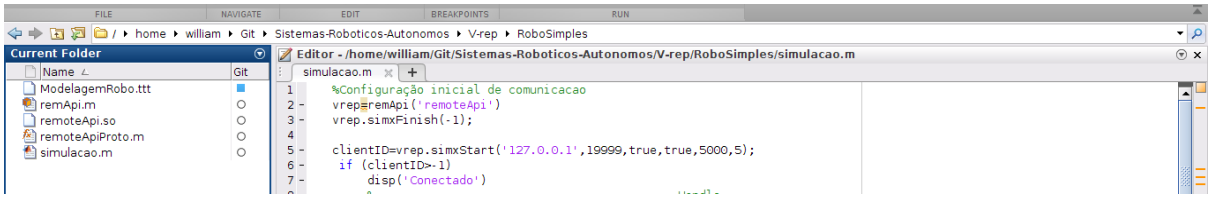

Figura 3: arquivos adicionados ao matlab

3.2 Programação

A programação que foi feita para teste do robô consiste basicamente em acionar os motores e quando o sensor de ultrassom, localizado na parte frontal do robô, identificar algum obstáculo a 0.3m de distância, ele começará a se movimentar para o lado.

Como foi utilizado o matlab para fazer a programação do robô, foram utilizados comandos específicos da biblioteca Remote API functions. Na documentação da biblioteca podemos encontrar as funções já implementadas de movimentação, por exemplo, e como fazer utilizá-la para movimentar o robô.

O código abaixo é utilizado para realizar a configuração da comunicação do robô.

```
% Configurao  inicial de  comunicao
vrep=remApi('remoteApi')
vrep.simxFinish(-1);

clientID=vrep.simxStart('127.0.0.1',19999,true,true,5000,5);
 if (clientID>-1)
     disp('Conectado')
```

Nesse segundo trecho de código, inicializam-se as variáveis globais, as quais irão se referir à um dos objetos do robô. A variável "left Motor", por exemplo, comunicar-se-á com o objeto LeftMotor do V-REP.

```
[returnCode,left_Motor]=vrep.simxGetObjectHandle(clientID,'LeftMotor',vrep.simx_opmode_
blocking)
[returnCode,right_Motor]=vrep.simxGetObjectHandle(clientID,'RigthMotor',vrep.simx_opmode_
```

```
blocking)
[returnCode,front_Sensor]=vrep.simxGetObjectHandle(clientID,'FrontUS',vrep.simx_opmode_
blocking) %Primeira chamada
[returnCode,carro]=
    vrep.simxGetObjectHandle(clientID,'RobotBase',vrep.simx_opmode_blocking)
```

Finalmente, está função é responsável por dizer qual será a velocidade do robô e pela aquisição da posição (x,y,z) ao longo do percurso. Neste caso, coloca-se a velocidade desejada, que será incrementada por 0.1 ao longo do tempo. Através da função "vrep.simxGetObjectPosition"se obtém os pontos (x,y,z) no MATLAB e guardados em vetores para serem plotados em tempo real conforme o robô percorre o caminho, além de realizar a leitura do sensor ultrassônico.

```
        %Incremento da velocidade
        [returnCode]=vrep.simxSetJointTargetVelocity(clientID,left_Motor,vel,vrep.simx_opmoc
        [returnCode]=vrep.simxSetJointTargetVelocity(clientID,right_Motor,vel,vrep.simx_opmo
        vel = vel + 0.1;
        %Parametro para pegar a posicao
        [returnCode,position]=vrep.simxGetObjectPosition(clientID,carro,-1,vrep.simx_opmode_
            %(Retorna X,Y,Z)
        %Pega a distncia pelo sensor
        [returnCode,detectionState,detectedPoint,~,~]=vrep.simxReadProximitySensor(clientID,
            %demais chamadas
        %disp(norm(position));

        %Salvar valores de x,y,z e teta nas variaveis
        x(i) = position(1);
        y(i) = position(2);
        z(i) = position(3);
        %plot em tempo real
      hold on,plot(x,'b')
      hold on,plot(y,'g')
      hold on,plot(z,'r')
      legend(' x',' y',' z');
```

O restante do código é apenas lógica para movimentação e leitura do sensor. No final de todo o código, para finalizar a comunicação do MATLAB com o Vrep, devemos inserir o comando vrep.delete().

4 Conclusão

O V-REP oferece API's remotas que permitem o controle da simulação e do simulador em si por uma aplicação externa ou, até mesmo, por um computador remoto. Neste trabalho explicitamos a comunicação com o MATLAB, apenas uma das possibilidades. O V-REP suporta comunicação com Python, C/C++, Java, Octave e Lua.

5 Referências

Os códigos utilizados neste relatório encontram-se Disponíveis em: <https://github.com/willcribeiro/Sistemas-Roboticos-Autonomos/tree/master/V-rep/Projeto%201>. Acesso em: 18 de maio de 2021. <https://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsMatlab Acesso em: 18 de maio de 2021.