

Robot Training - Projeto Ararajuba

Breno Pinheiro de Meneses
Gabriel Henrique Vasconcelos da Silva
Marina Oliveira Batista

Universidade Federal de Campina Grande - UFCG
Centro de Engenharia Elétrica e Informática - CEEI
Departamento de Engenharia Elétrica - DEE

10 de junho de 2022

- **Sensores**

- Caracterizar o modelo dos plugins utilizados na simulação dos sensores
- Especificar os sensores físicos a serem usados a priori
- Implementação do Sensor Tag no ROS

IMU plugin - libgazebo_ros_imu_sensor.so

- O modelo utilizado para geração de dados do IMU por meio do gazebo, é a obtenção de dados Ground Truth somados com erro gaussiano

Modelo IMU

```
orientation.x = orientation.X() + GaussianKernel(0,gaussian_noise);  
linear_acceleration.x = accelerometer_data.X() + GaussianKernel(0,gaussian_noise);  
angular_velocity.x = gyroscope_data.X() + GaussianKernel(0,gaussian_noise);
```

Link: Plugin IMU

- A geração do erro gaussiano é feita da seguinte forma:
 - É realizada uma amostragem de dois números aleatórios

$$\sigma_1 = rand(seed)$$

$$\sigma_2 = rand(seed)$$

- Estes são aplicados em uma transformação de Box-Müller para amostrar um número aleatório normalmente distribuído

$$Z_0 = \sqrt{-2 \log \sigma_1} \cos 2\pi \sigma_2$$

- Após isso é feita a normalização para que possua a média e variância definidas

$$Z = \mu_0 + Z_0 \sigma$$

Link: Box_Müller

Sensor Tag - CC2650

- Conexão: Bluetooth low energy (BLE)
- 10 sensores inclusos
- IMU - unidades:
 - Accel: G
 - Gyro: $^{\circ}/s$



Figura 1: Sensor Tag - CC2650

LiDAR plugin - libgazebo_ros_laser.so

- O plugin utiliza sensores do tipo ray distribuídos para o cálculo das distâncias e intensidades, limitadas a máxima distância e intensidade especificadas
- Caso o número de raios simulados for menor que o número de leituras desejadas é feita a interpolação de duas medições

Link: Plugin LiDAR

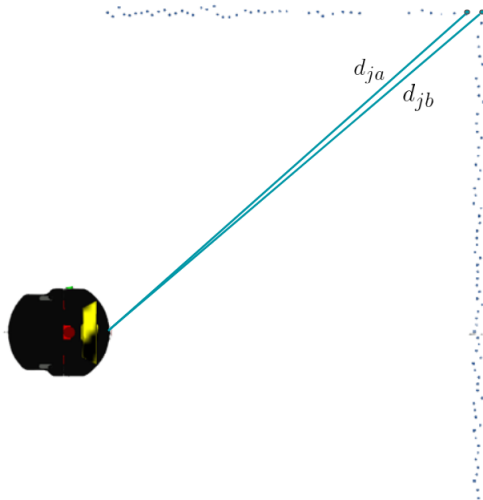


Figura 2: Princípio de funcionamento do plugin laser

LiDAR plugin - libgazebo_ros_laser.so

- Para publicação de dados do LiDAR é criado um vetor de distância e intensidade das leituras dos sensores pontuais distribuídos somadas ao erro gaussiano

Modelo LiDAR

```
ranges.push_back(std::min(r + minRange + this->GaussianKernel(0,this->gaussianNoise), maxRange));  
intensities.push_back(std::max(this->hokuyoMinIntensity,intensity + this->GaussianKernel(0,this->gaussianNoise)));
```

Link: Plugin LiDAR

Neato XV LiDAR

- Alcance: 10cm até 5m
- Ângulo: 360°
- Comunicação Serial a 3.3V
- Taxa de Transmissão: 115200bps



Figura 3: Neato XV LiDAR

Camera plugin - libgazebo_ros_openni_kinect.so

- O plugin utiliza o modelo da Câmera de Pinhole



- **GazeboRosCameraUtils:** Define todos os parâmetros da câmera e de distorção baseado no modelo da câmera de Pinhole utilizado no OpenCV.

Links: OpenCV - Modelo Pinhole
Plugin Openni Kinect

Camera plugin - libgazebo_ros_openni_kinect.so

Parâmetros definidos no gazebo

- Tipo de câmera
- Coeficientes de Distorção
- Distância Focal
- Resolução

Microsoft Kinect - 1414

• Fluxos de dados

- 320×240 16-bit depth 30 frames/s.
- 640×480 32-bit color 30 frames/s.



Figura 4: Microsoft Kinect - 1414

- Implementação do Sensor Tag

- **Fusão de Sensores**

- Observar o funcionamento interno do `robot_ekf_pose` e o `laser_scan_matcher`
- Testes para aquisição de dados dos sensores físicos

Cronograma

- Início da Semana 1: 12 de maio de 2022
- Final da Semana 9: 14 de julho de 2022

Etapas	Semanas								
	1	2	3	4	5	6	7	8	9
1	x								
2		x	x						
3			x	x	x				
4					x	x			
5						x	x		
6							x	x	x

Obrigado!

Breno Pinheiro de Meneses
Gabriel Henrique Vasconcelos da Silva
Marina Oliveira Batista

Universidade Federal de Campina Grande - UFCG
Centro de Engenharia Elétrica e Informática - CEEI
Departamento de Engenharia Elétrica - DEE

10 de junho de 2022

`breno.meneses@ee.ufcg.edu.br`
`gabriel.vasconcelos@ee.ufcg.edu.br`
`marina.batista@ee.ufcg.edu.br`

