

Insper

Robótica Computacional

Comportamento, Laser e Odometria

Insper

Comportamento

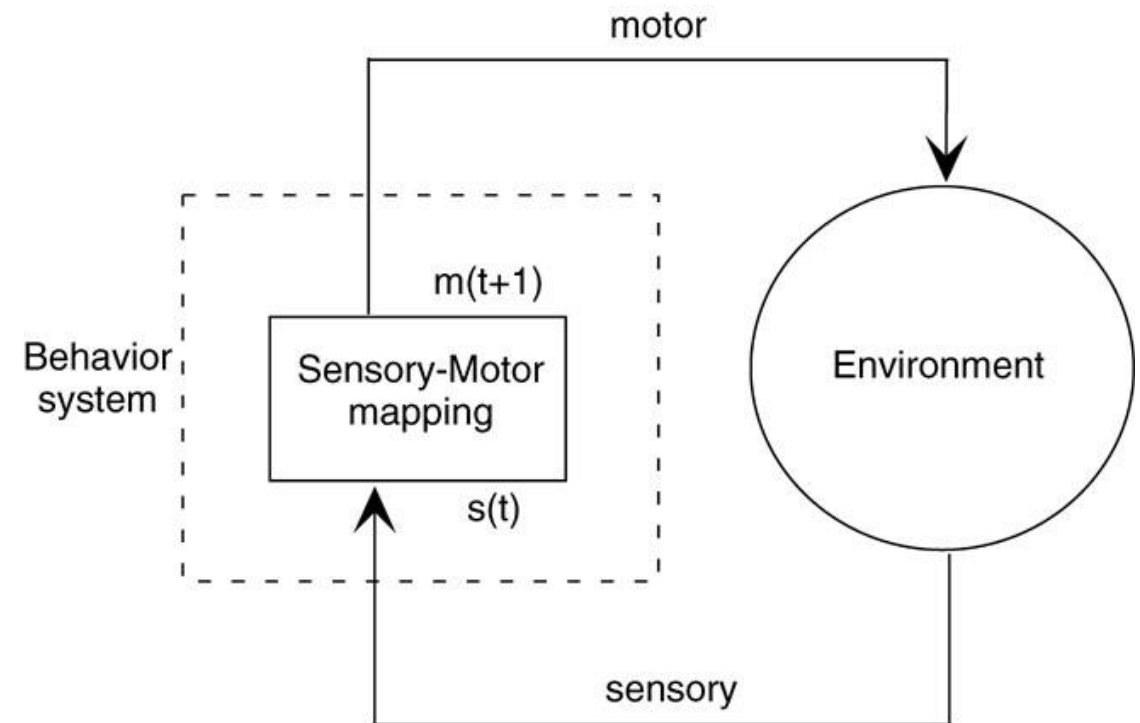
Como decidir o comportamento do robô?

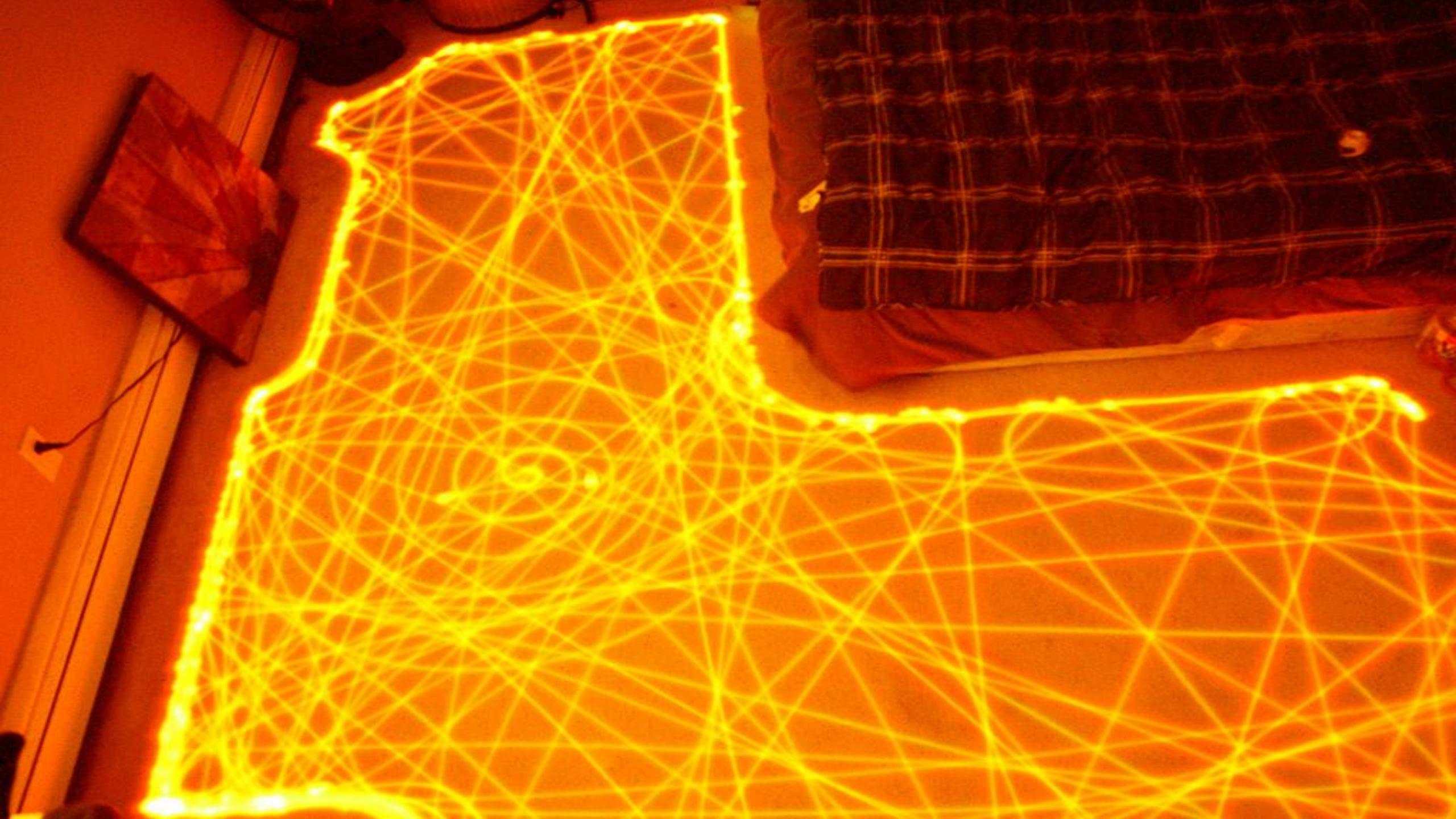
Na primeira aula discutimos o seguinte cenário. Qual ação o robô deveria tomar?



Estrutura de controle de robô

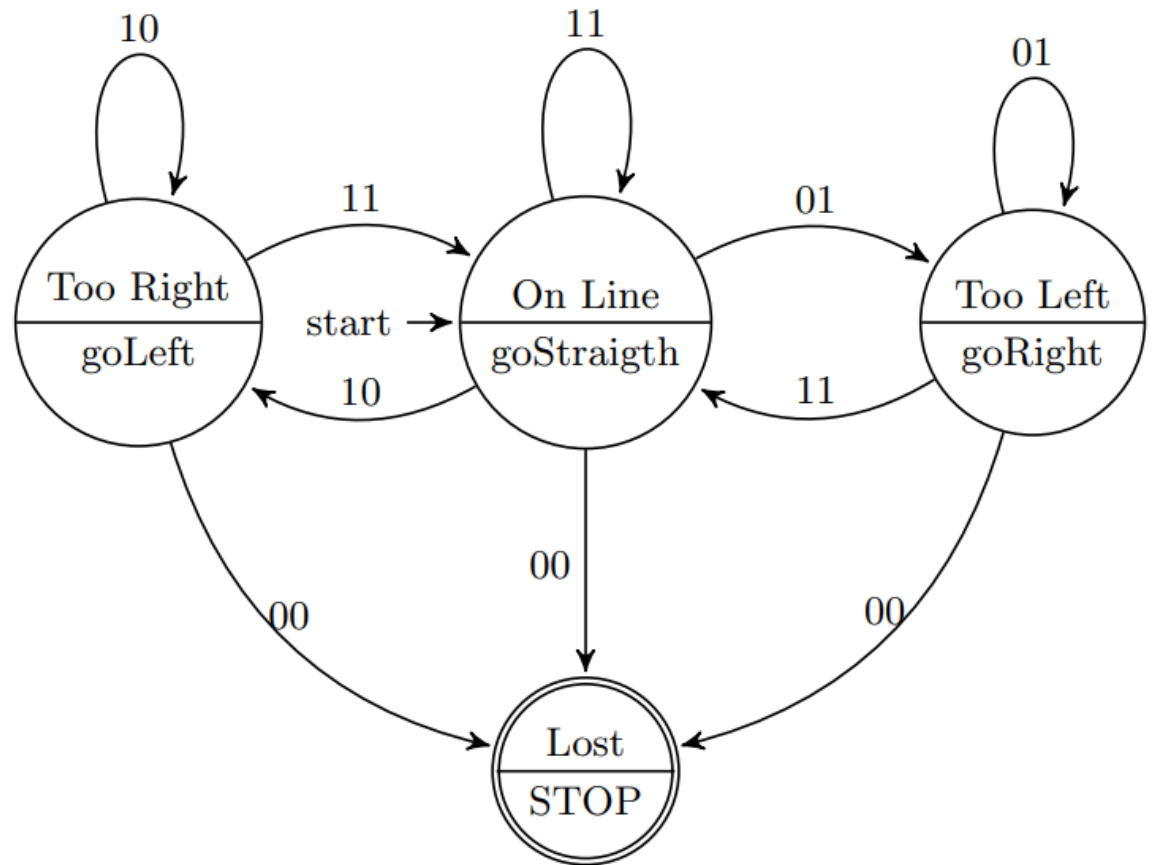
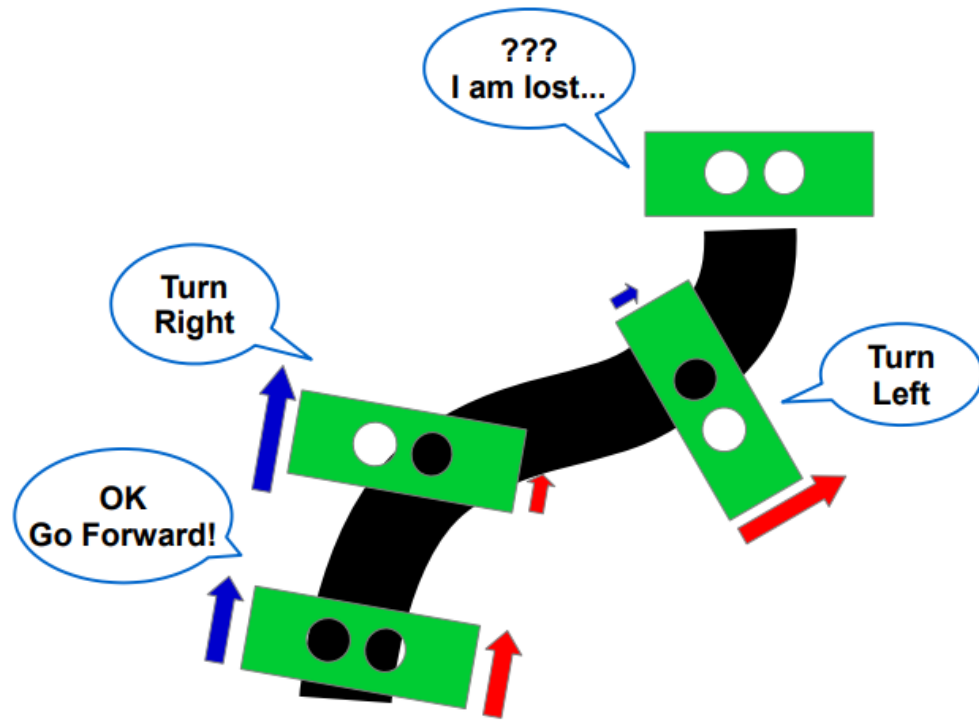
- Um robô autônomo vê o ambiente ao redor e depois executa uma ação.
- Dependendo do seu comportamento, podemos ter diferentes resultados.





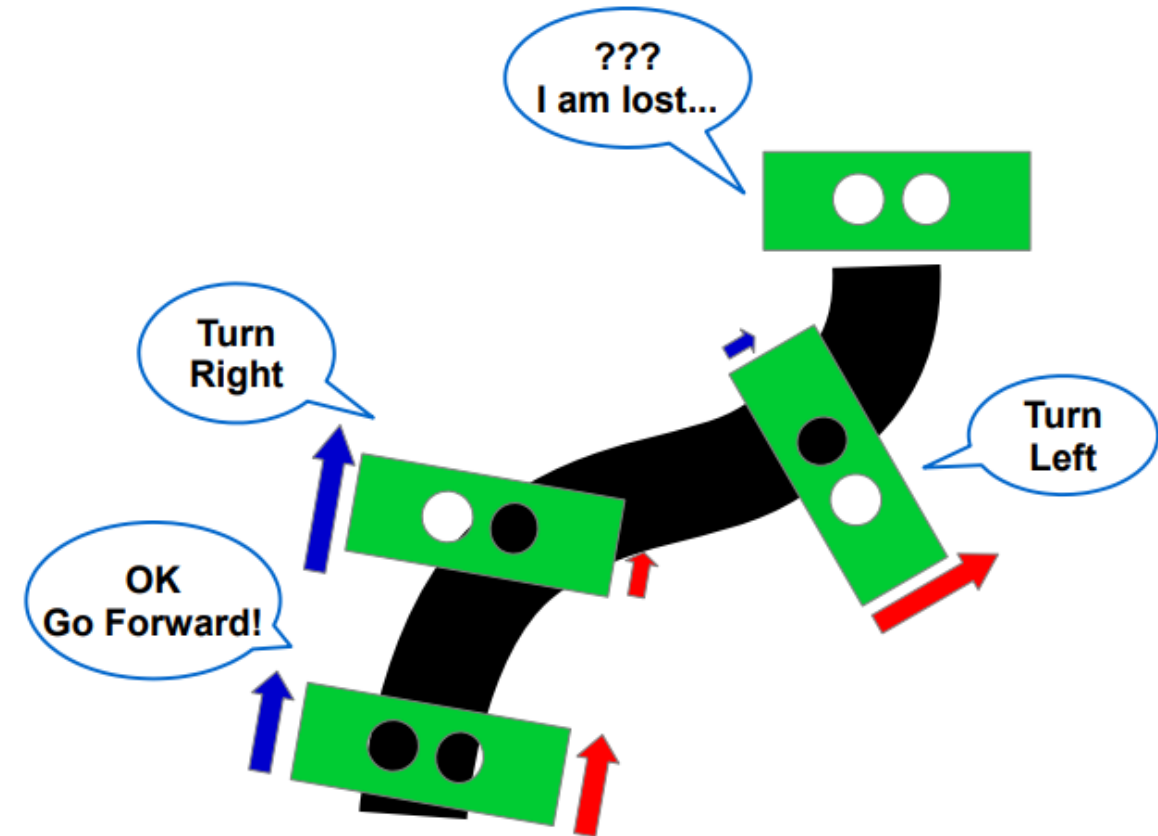


Máquina de Estado



Máquina de Estado

- Neste exemplo de um robô seguidor de linhas, ele tem 4 estados:
 - Virar a direita
 - Virar a esquerda
 - Linha reta
 - Parado
- Decisão é feita através do output do sensor IR



Máquina de Estado

```
class Control():
    def __init__(self):
        self.robot_state = "procura" # Estado atual

        self.robot_machine = {
            "procura": self.procura,
            "aproxima": self.aproxima,
            "para": self.para
        }

    def procura(self) -> None:
        ...
        self.robot_state = "aproxima"

    def aproxima(self) -> None:
```

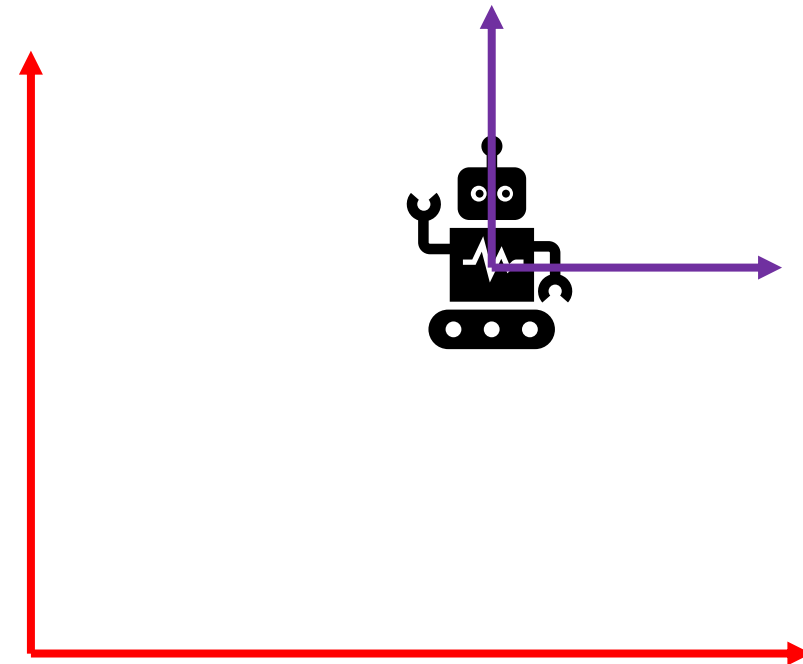
Insper

Odometria

Localização

Localização ajuda o robô a interagir melhor com o ambiente.

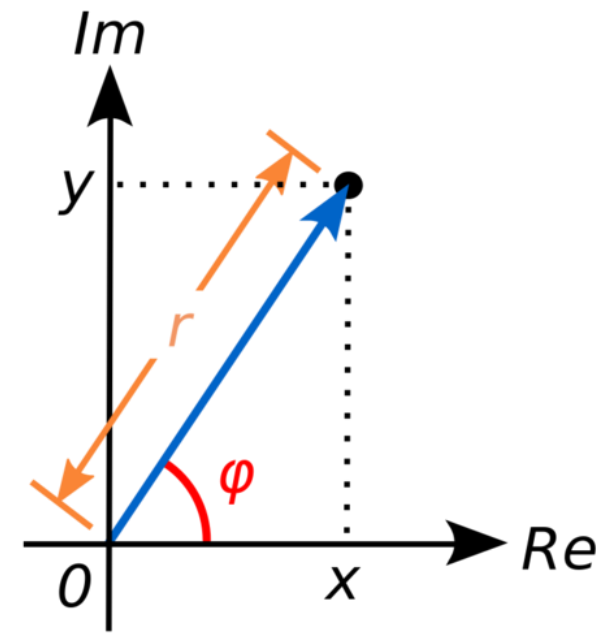
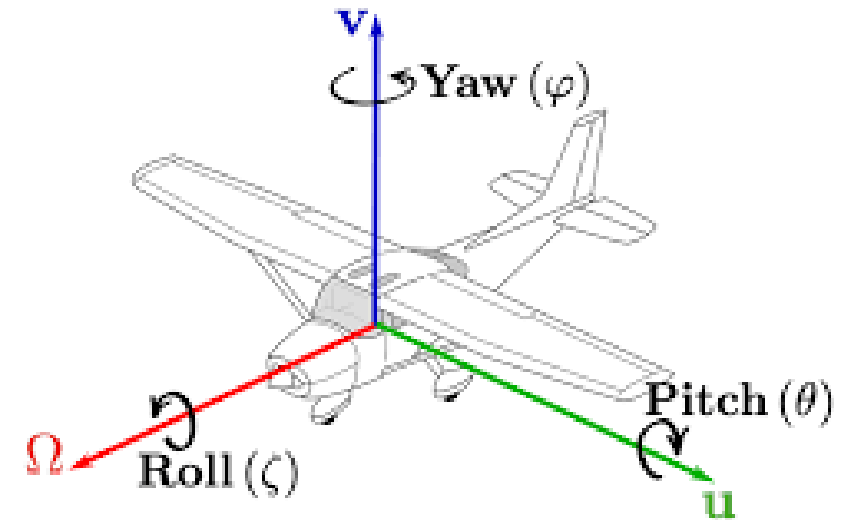
- O mapa consiste em uma vista superior do mundo e pode ser definido a partir de:
 - Coordenadas **Global**: Eixo fixo no mundo
 - Coordenadas **Local**: Eixo no robô
- Na ROS a odometria é dada como:
 - Robô Simulado: Localização absoluta.
 - Robô Real: Localização com base no *encoder* das rodas.



Localização

Além de localização, o robô tem uma orientação.

- Orientação podem ser expressas de duas formas:
 - Euler (3-upla)
 - Combinação de rotação:
 - $R = R_x * R_y * R_z$
 - Quaternions (4-upla)
 - Números imaginários



Odometria

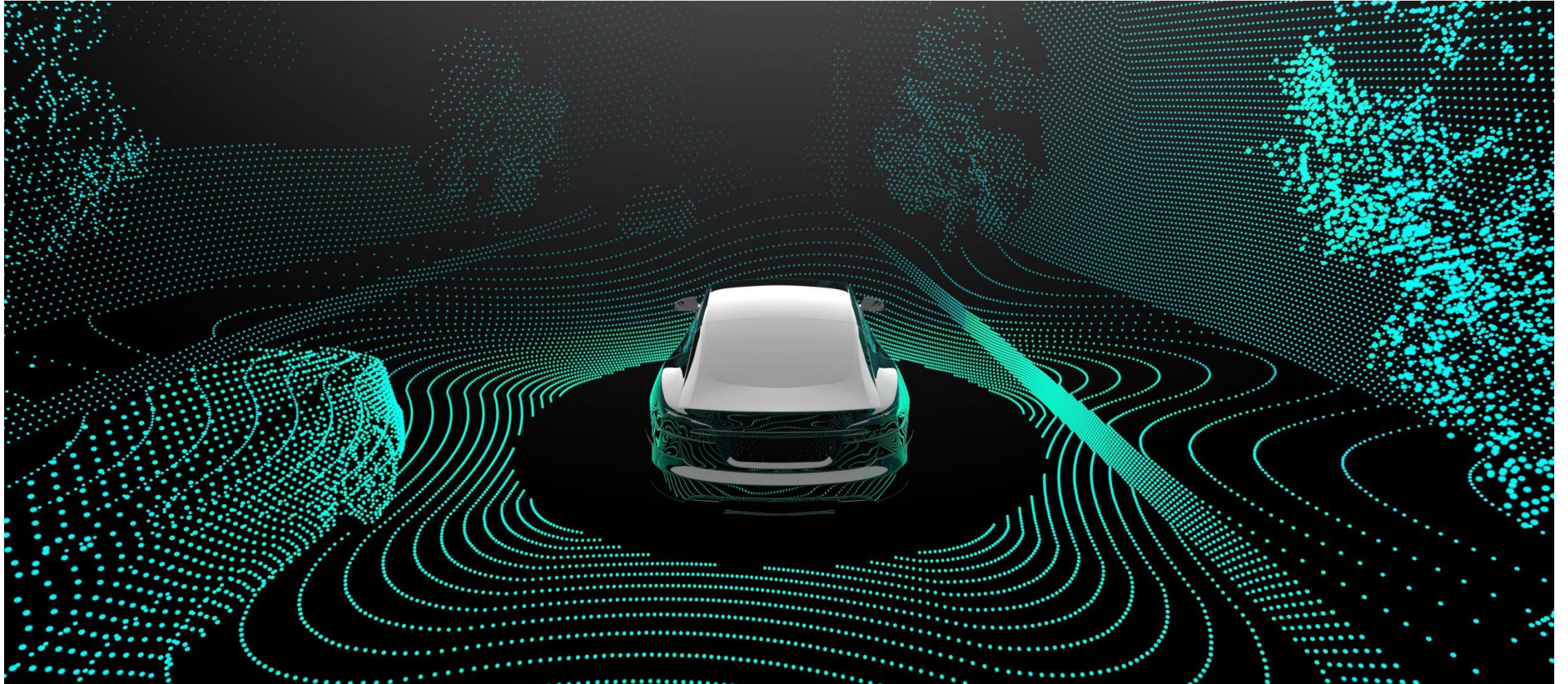
Tópico: /odom

```
def odom_callback(self, data: Odometry):  
    self.position = data.pose.pose.position  
  
    orientation_list = [data.pose.pose.orientation.x,  
                        data.pose.pose.orientation.y,  
                        data.pose.pose.orientation.z,  
                        data.pose.pose.orientation.w]  
  
    self.roll, self.pitch, self.yaw = euler_from_quaternion(orientation_list)
```

Insper

Laser

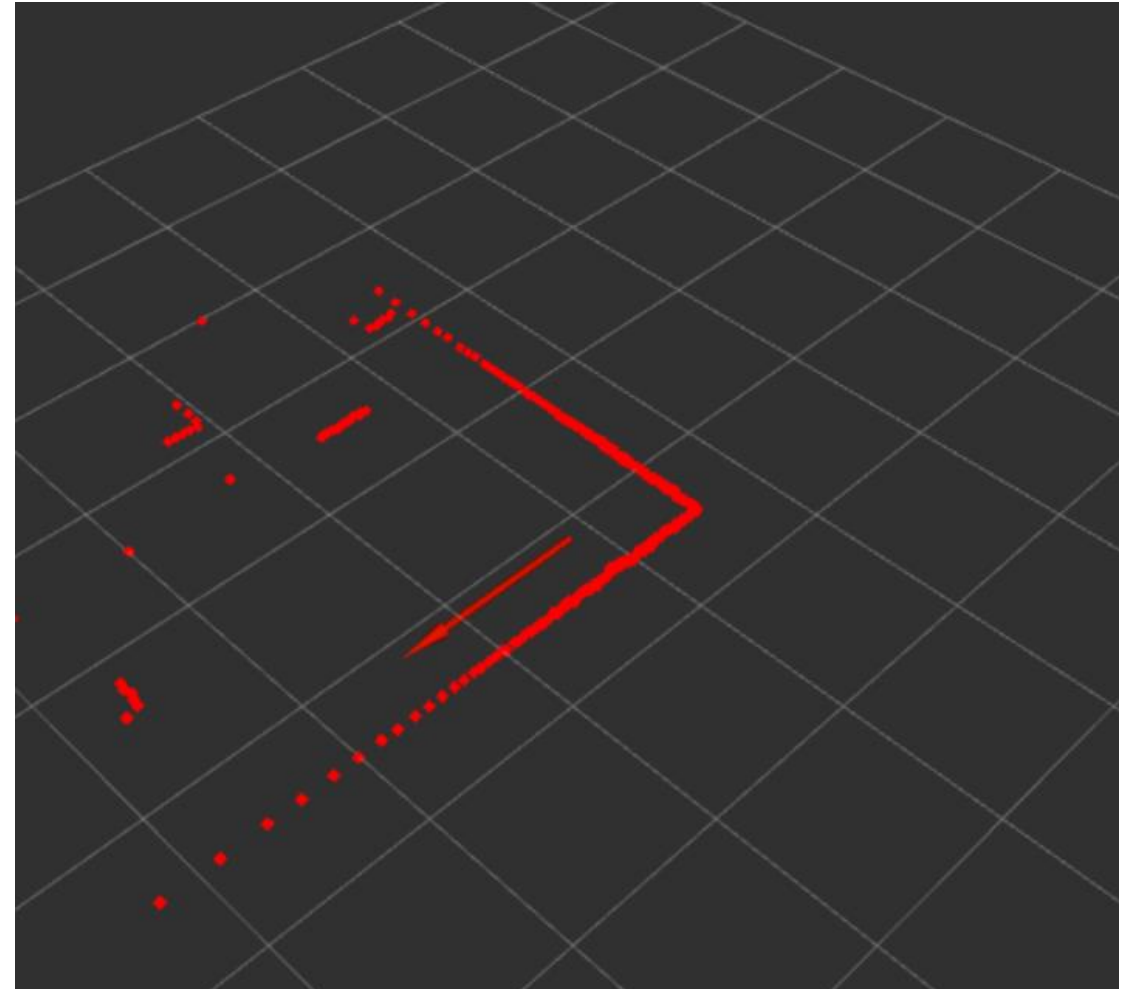
LIDAR



Laser – Robô

Durante a disciplina, vamos não vamos utilizar um sensor LIDAR, mas sim um sensor Laser 2D.

- **Precisão Angular:** Medições realizadas a cada 1° .
- **Alcance de Detecção:** Capaz de detectar objetos em um raio de até 3,5 metros.
- **Dados de Saída:** Lista detalhada de 360 pontos de medição.
- **Orientação:** O primeiro ponto de dados alinha-se com a parte frontal do robô.
 - Lembrando que o robô sempre olha para o eixo X.



Atividades

Agora estão prontos para seguir com as atividades individuais:

- Atividade 1 - Introdução ao conceito de máquina de estados.
- Atividade 2 – Estrutura básica de nó e controle.
- Atividade 3 – Entendendo o conceito de Pose e Odometria.
- Atividade 4 - Entendendo a leitura do sensor laser.
- Atividade 5 – Visualização de dados de sensor na ROS 2.