ROBOTICS LAB 8
Laser Range Finder

MUNADI SIAL

SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

Overview

This lab will focus on the following:

- Subscribing to the Laser Scan topic
- Acquiring data from the Laser Range Finder
- Wandering Robot
- Wall Following Robot

Overview

- We have looked at ROS communication between nodes via topics
- We have also looked at the actuating aspect of robotics (twist messages)

- In this lab, we will turn our attention to the **sensing** aspect of robotics
- Specifically, the focus will be on the Laser Range Finder which is a proximity sensor used widely for mapping, localization, obstacle avoidance and navigation purposes
- This lab will involve learning to use the laser data, then developing simple implementations for wandering and wall following

Sensors

Sensors are devices used to get useful information from the environment

Vision Sensor
CCD/CMOS camera

LASER Scanner
Measures distance from obstacle

SONAR
Measures distance from obstacle

Sensors

Sensors are devices used to get useful information from the environment

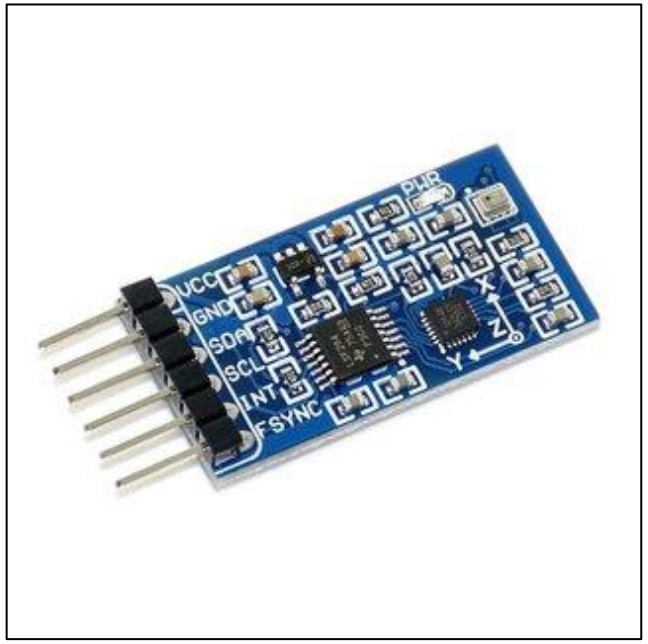
Encoder
Measures angular position

Tachometer
Measures angular velocity

GPS
Measures location in outdoor environments

Sensors

Sensors are devices used to get useful information from the environment



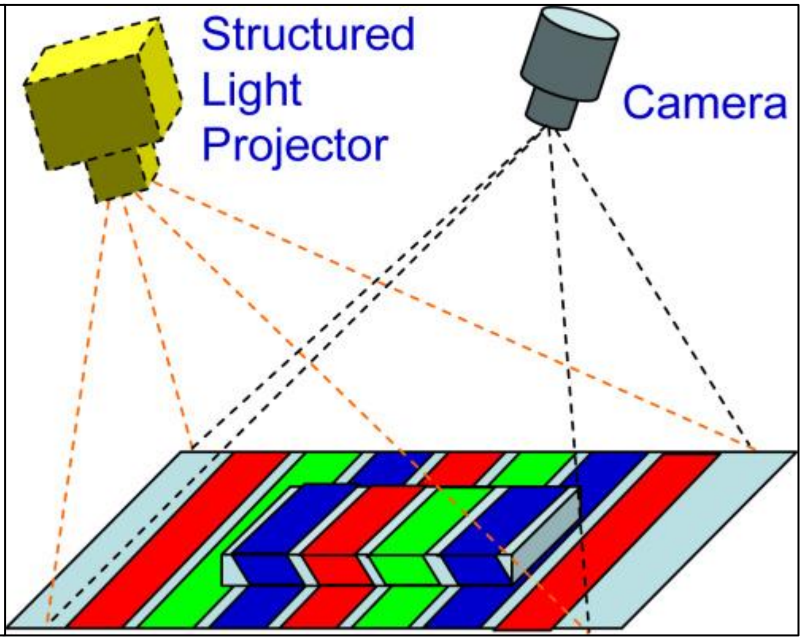
IMU
Accelerometer and gyroscope combined (compass is also included sometimes)

Accelerometer
Measures linear acceleration

Rate Gyroscope
Measures angular velocity of the body to which it is attached

Sensors

Sensors are devices used to get useful information from the environment

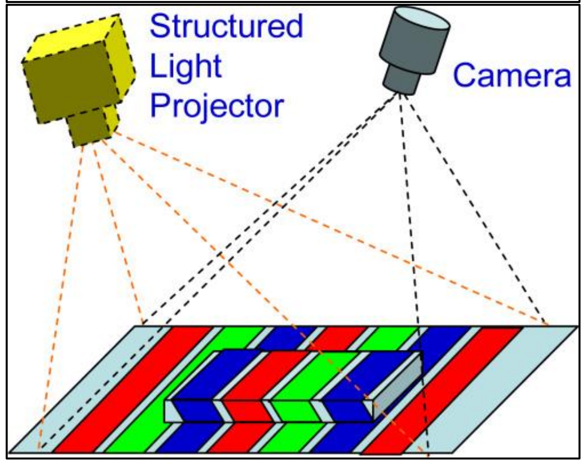


Tactile Sensor
Used to sense touch by detecting bumps

Kinect Sensor
Projects "structured light" on the scene and the change in pattern is used to obtain information of the scene geometry

Range Sensors

- A Range Sensor is a device that measures the distance to nearby, external objects
- Examples of range sensors include
  - Sonars
  - Radars
  - Laser Range Finders
  - Structured Light



Kinect Sensor
It projects "structured light" on the scene

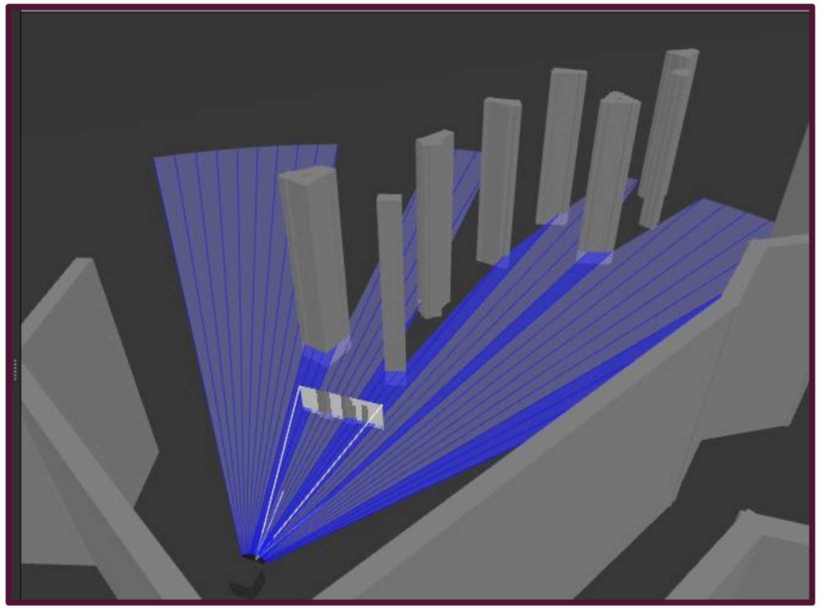
Sonar Sensor

Laser Range Finder

Laser Range Finder

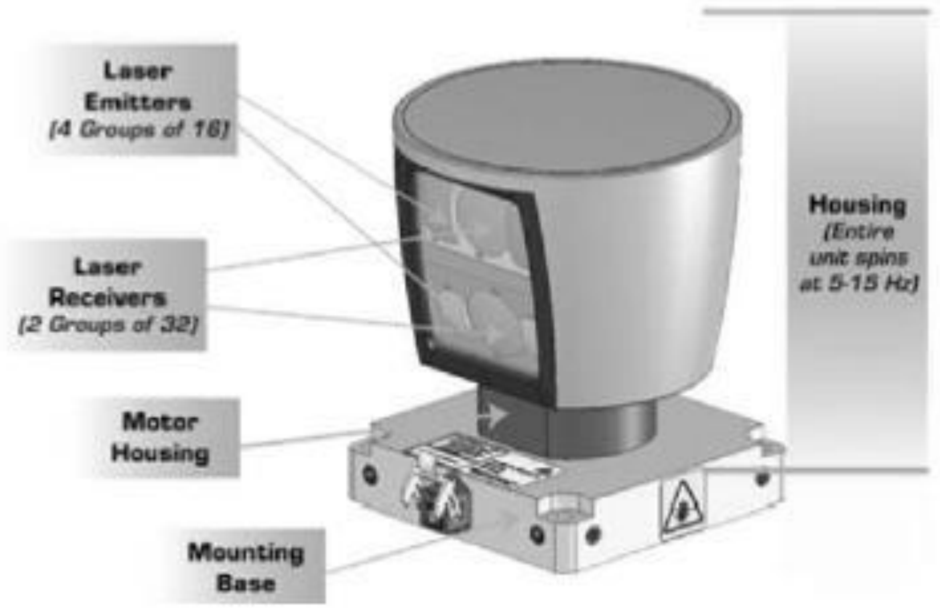
- A Laser Range Finder measures the distance from an object by firing numerous lasers and then obtaining the reflected response

- Laser range finders have some advantages compared to other range sensors
  - Larger distance range
  - Larger field of view (even up to 360° around)
  - Higher frequency of measurements
  - Higher accuracy in measurements



Laser Range Finder

- For indoor applications (where GPS is not practical), laser range finders are used for mapping, localization and obstacle avoidance
- For outdoor applications, laser range finders are used in autonomous cars to detect pedestrians and traffic. Due to the high speed of such cars, the laser sensors must have a large range, e.g. the Velodyne HDL-64E (shown) has a range of 50 m and an accuracy within 2 cm


Laser Emitters (4 Groups of 16)
Laser Receivers (2 Groups of 32)
Motor Housing
Mounting Base
Housing (Entire unit spins at 5-15 Hz)

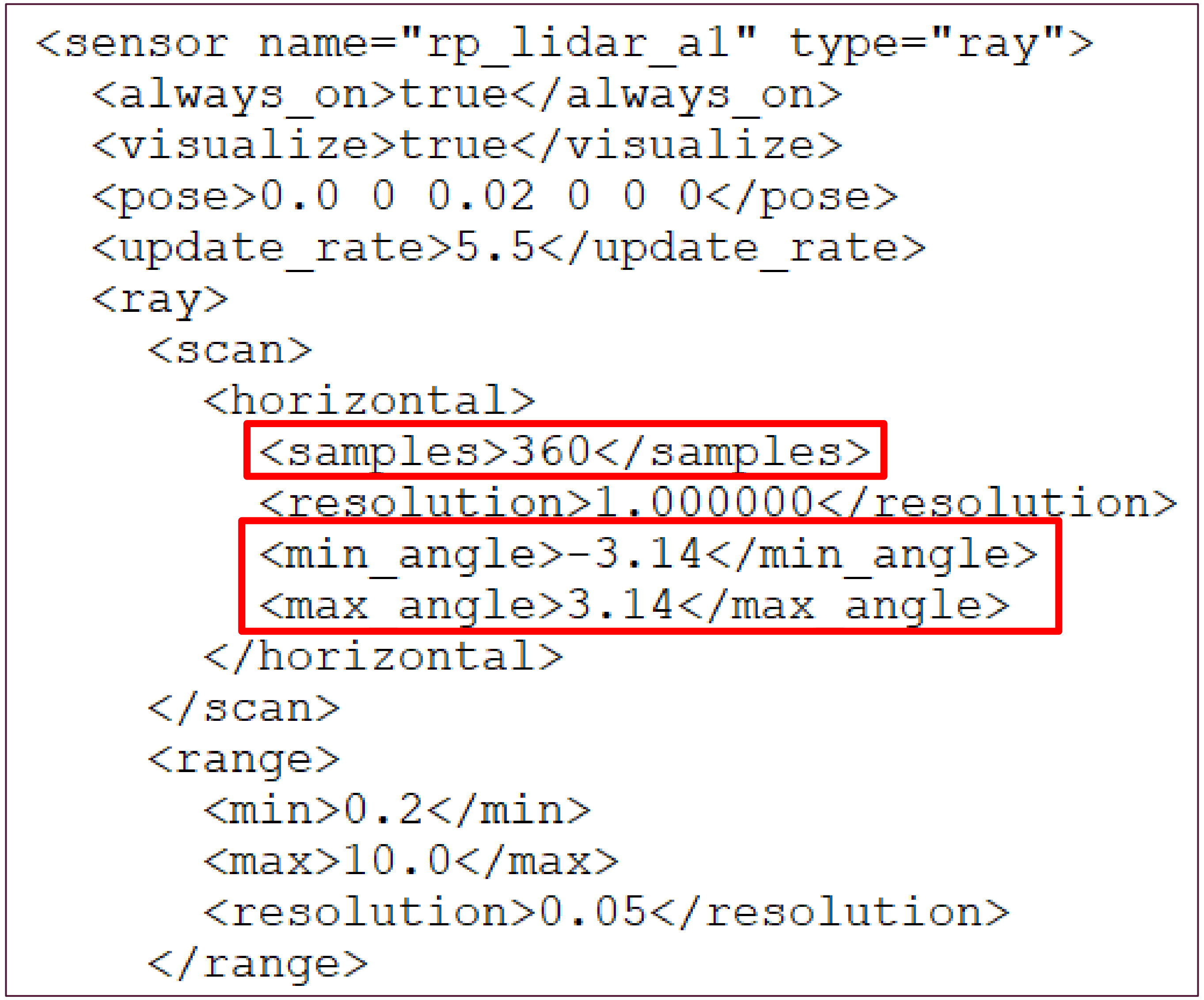
Limitations

Like any other sensor, laser range finder has its own limitations:

- **Measurement noise:** Sensor readings are subjected to noise due to limited resolution, atmospheric effects etc
- **Specular reflections:** Sensor readings are missed when the fired rays bounce off of objects at varied angles
- **Ambient light:** Some laser scanners fail to measure objects in the presence of bright sunlight
- **Transparent objects:** Laser scans will give much larger distances if the fired lasers pass through the object

Laser Modification

- To modify the laser in the robot, the values in the model.sdf file need to be changed
- The sdf file contains XML code for the geometry of the robot and its individual parts (including the laser)
- We will alter the values for the samples, min_angle and max_angle tags in the sdf file



```
<sensor name="rp_lidar_a1" type="ray">
  <always_on>true</always_on>
  <visualize>true</visualize>
  <pose>0.0 0 0.02 0 0 0</pose>
  <update_rate>5.5</update_rate>
  <ray>
    <scan>
      <horizontal>
        <samples>360</samples>
        <resolution>1.000000</resolution>
        <min_angle>-3.14</min_angle>
        <max_angle>3.14</max_angle>
      </horizontal>
    </scan>
    <range>
      <min>0.2</min>
      <max>10.0</max>
      <resolution>0.05</resolution>
    </range>
```

Message Libraries

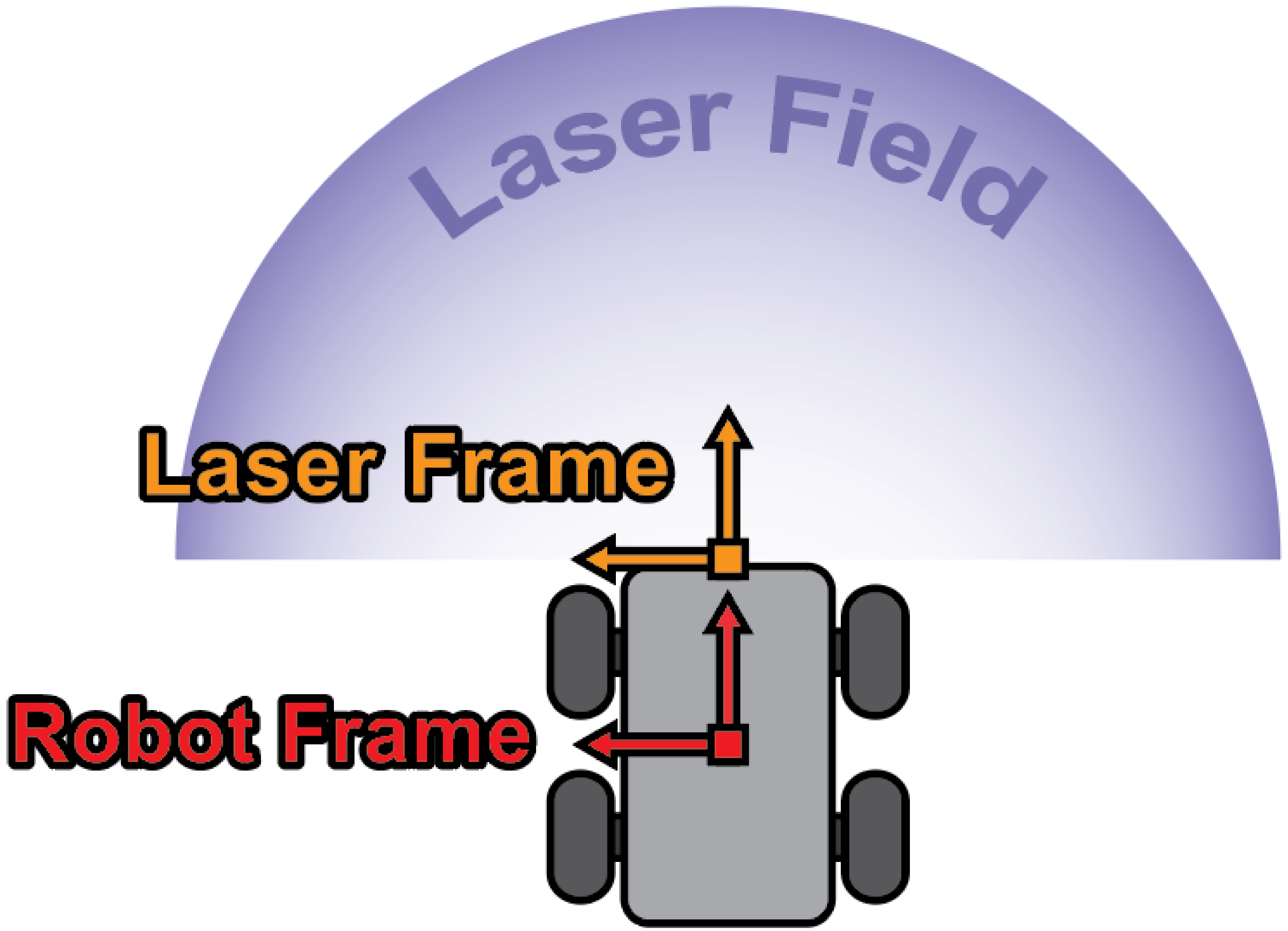
- The **std_msgs** library contains message types for standard messages:
  - Int32
  - Float32
  - String

- The **geometry_msgs** library contains message types for geometric data:
  - Pose
  - Point
  - Twist
  - Quaternion

- The **sensor_msgs** library contains message types for sensor data:
  - Image
  - $\boxed{\text{LaserScan}}$
  - PointCloud
  - MagneticFIeld

Laser Scan in ROS

- A robot with a laser field of 180° span is shown
- The origin of the Laser Frame is placed at the point where the lasers emerge.
- In the robot, the lasers are emitted from the very front of the robot's body

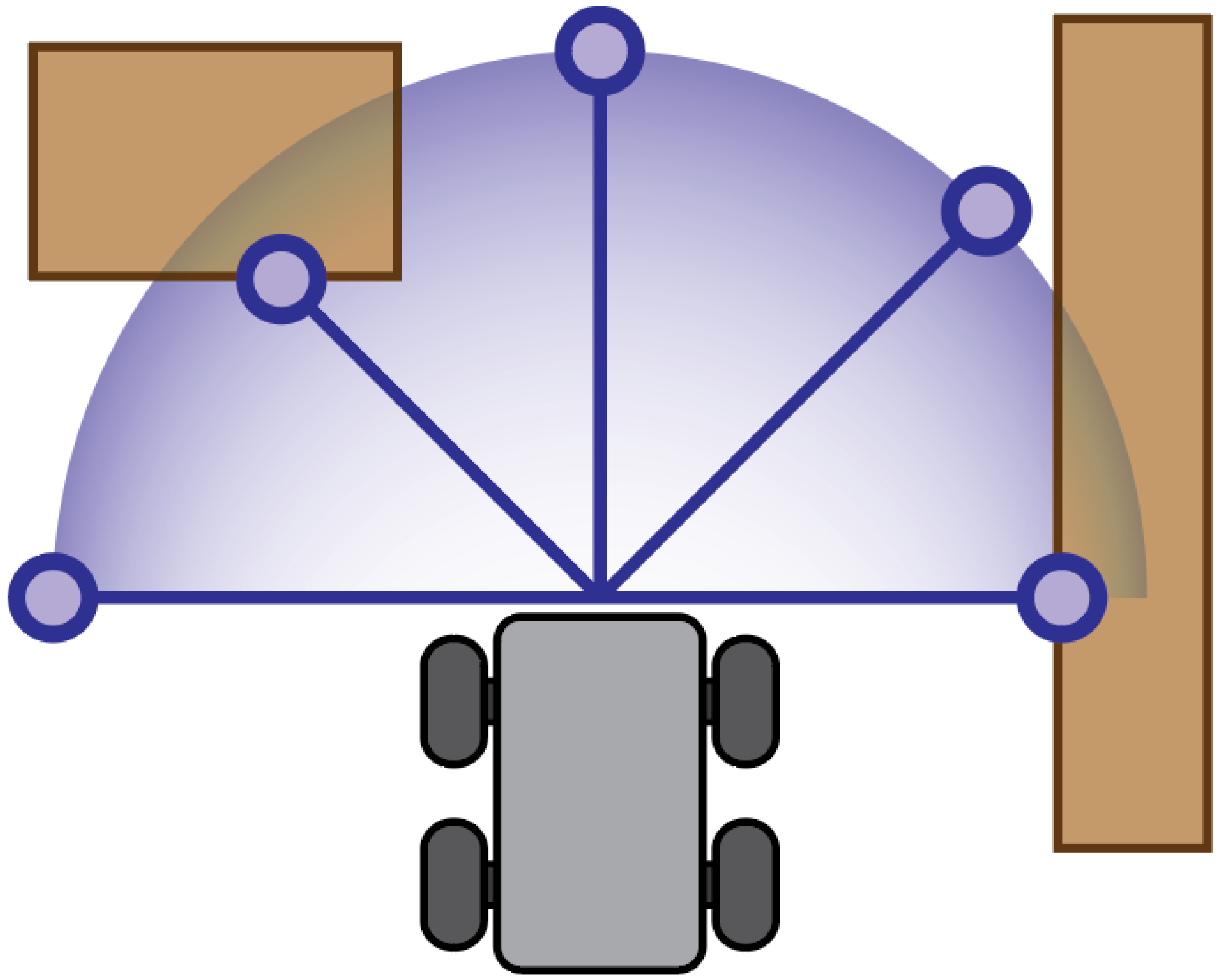


Laser Scan in ROS

Consider a robot with a laser spanning 180° and 5 laser samples:

- Left
- Front-left
- Front
- Front-right
- Right

The obstacles are shown to be hit by 2 lasers (front-left and right)

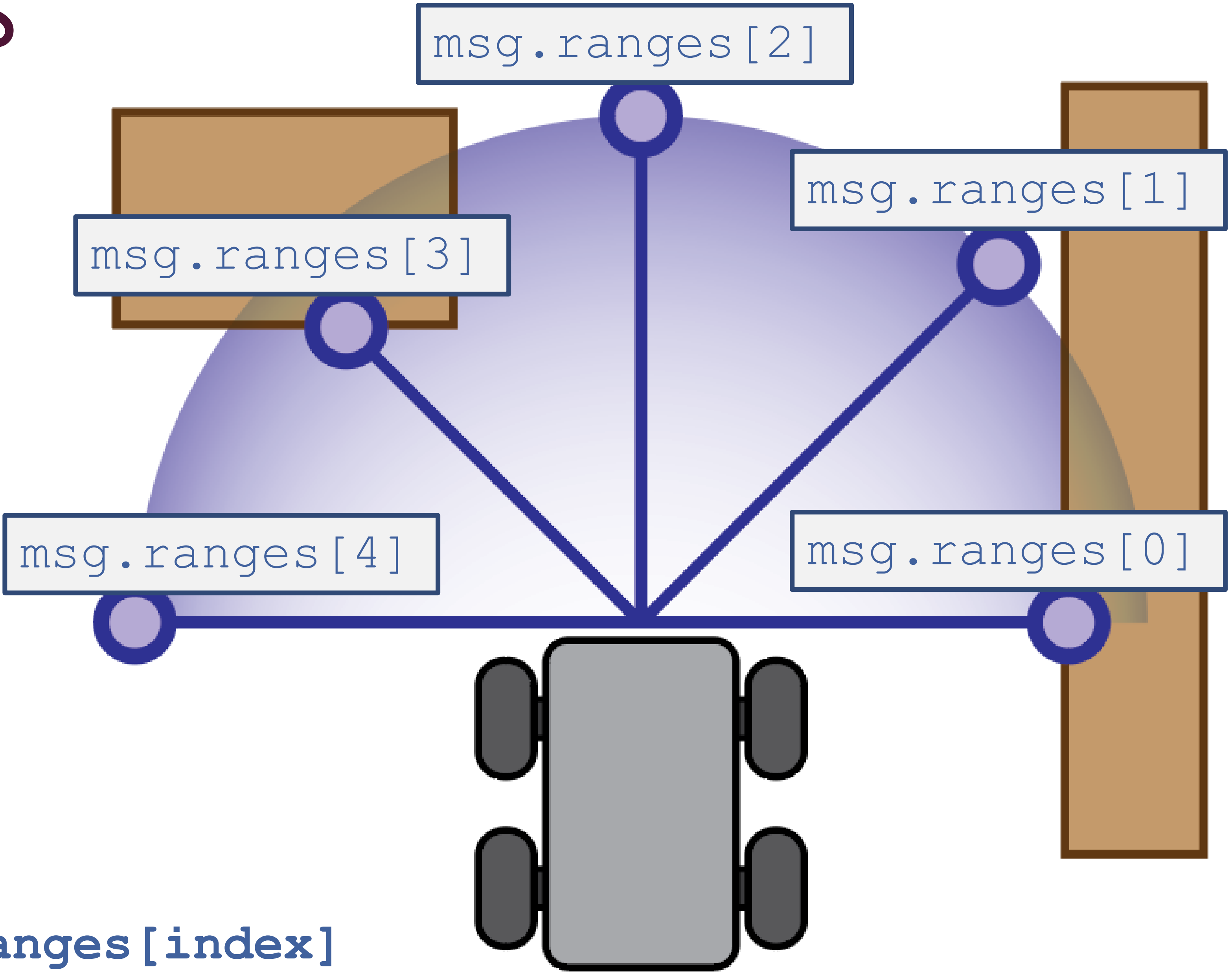


Laser Scan in ROS

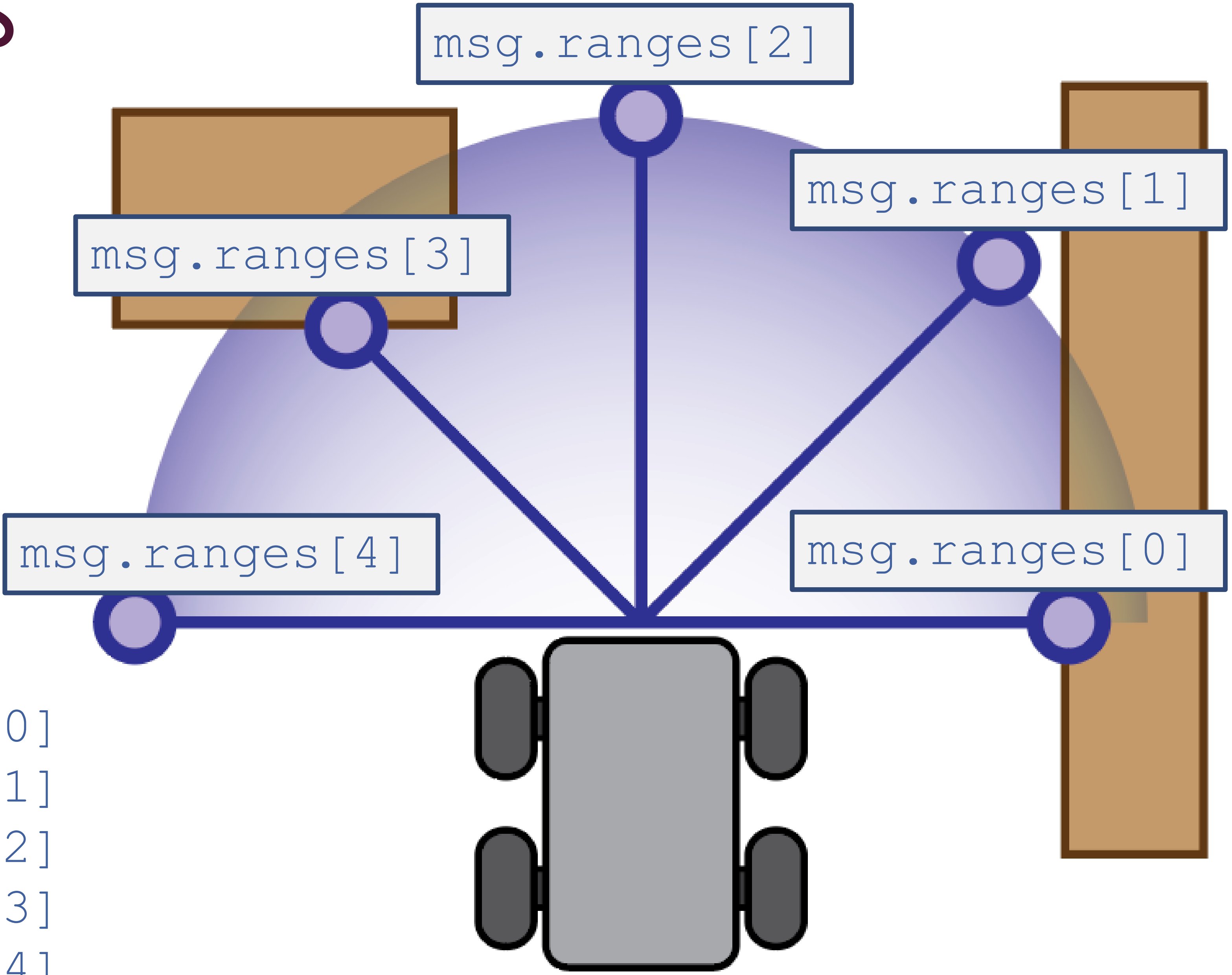
To get laser data, we create a subscriber node and subscribe to the **'scan'** topic which receives messages of type **LaserScan** (sensor_msgs)

If the message instance in the callback is *msg*, then the distance values of a laser sample can be obtained by the **ranges** attribute of LaserScan

```
range_value = msg.ranges[index]
```



Laser Scan in ROS



For the 5 laser samples, we can get the data by:

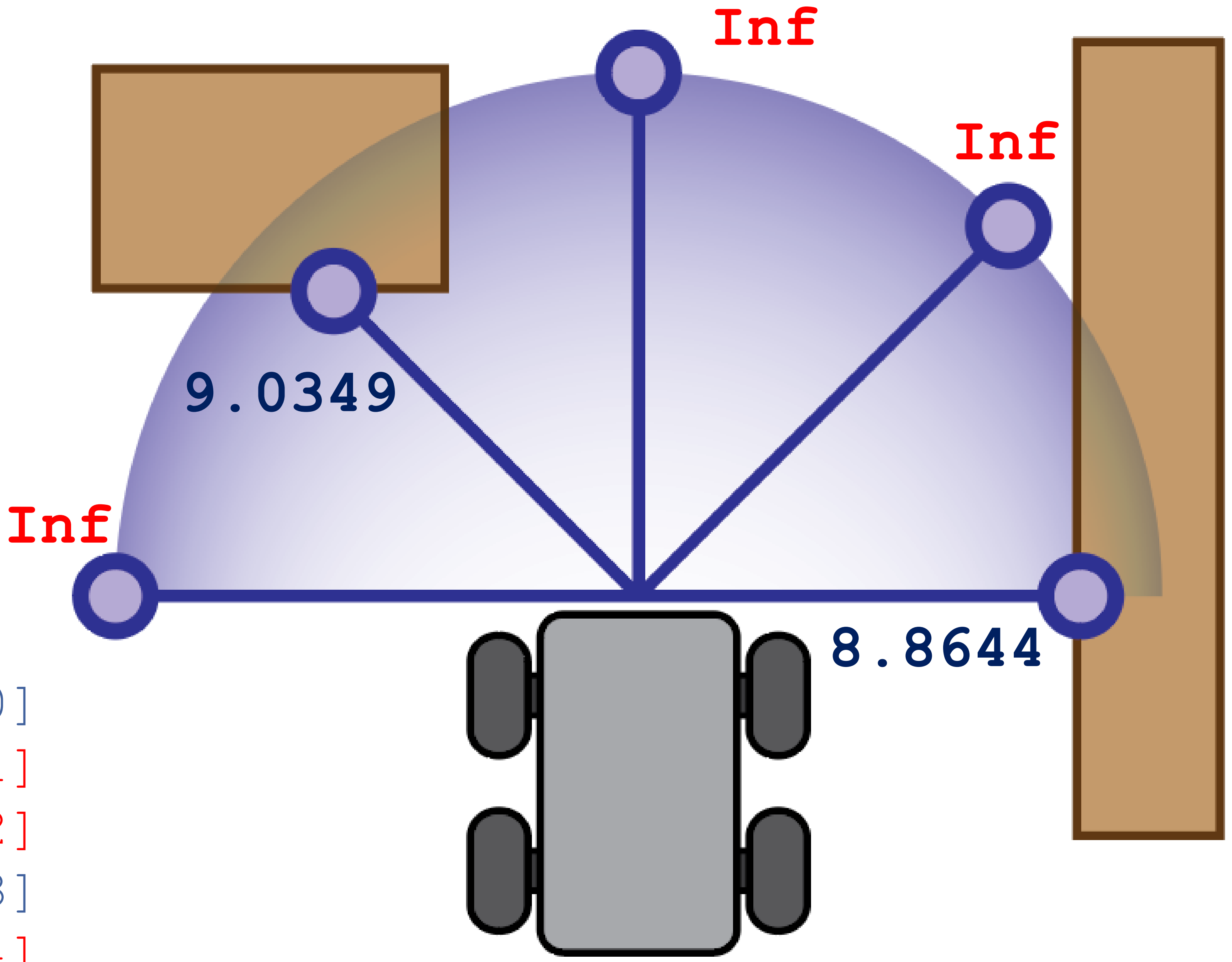
```
range_R  = msg.ranges[0]
range_FR = msg.ranges[1]
range_F  = msg.ranges[2]
range_FL = msg.ranges[3]
range_L  = msg.ranges[4]
```

Laser Scan in ROS

The 2 lasers that hit the obstacles give out proper values for the distance

The 3 lasers that do not hit any obstacle give the out of range values; infinity
Such values pose problems mathematically

```
range_R  = msg.ranges[0]
range_FR = msg.ranges[1]
range_F  = msg.ranges[2]
range_FL = msg.ranges[3]
range_L  = msg.ranges[4]
```

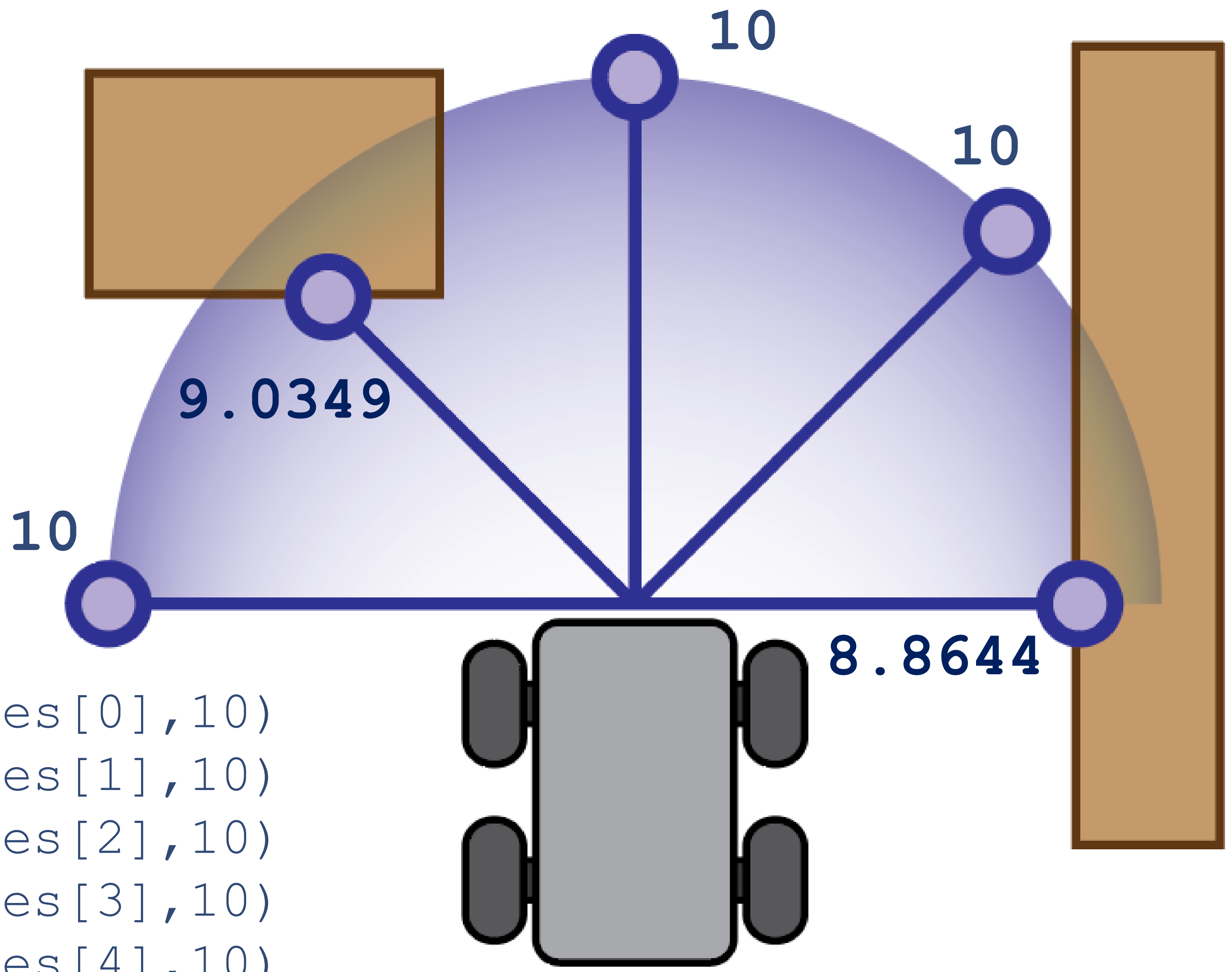


Laser Scan in ROS

To solve the problem of infinite values, we assign the max measurable value (laser range) in place of the infinite value

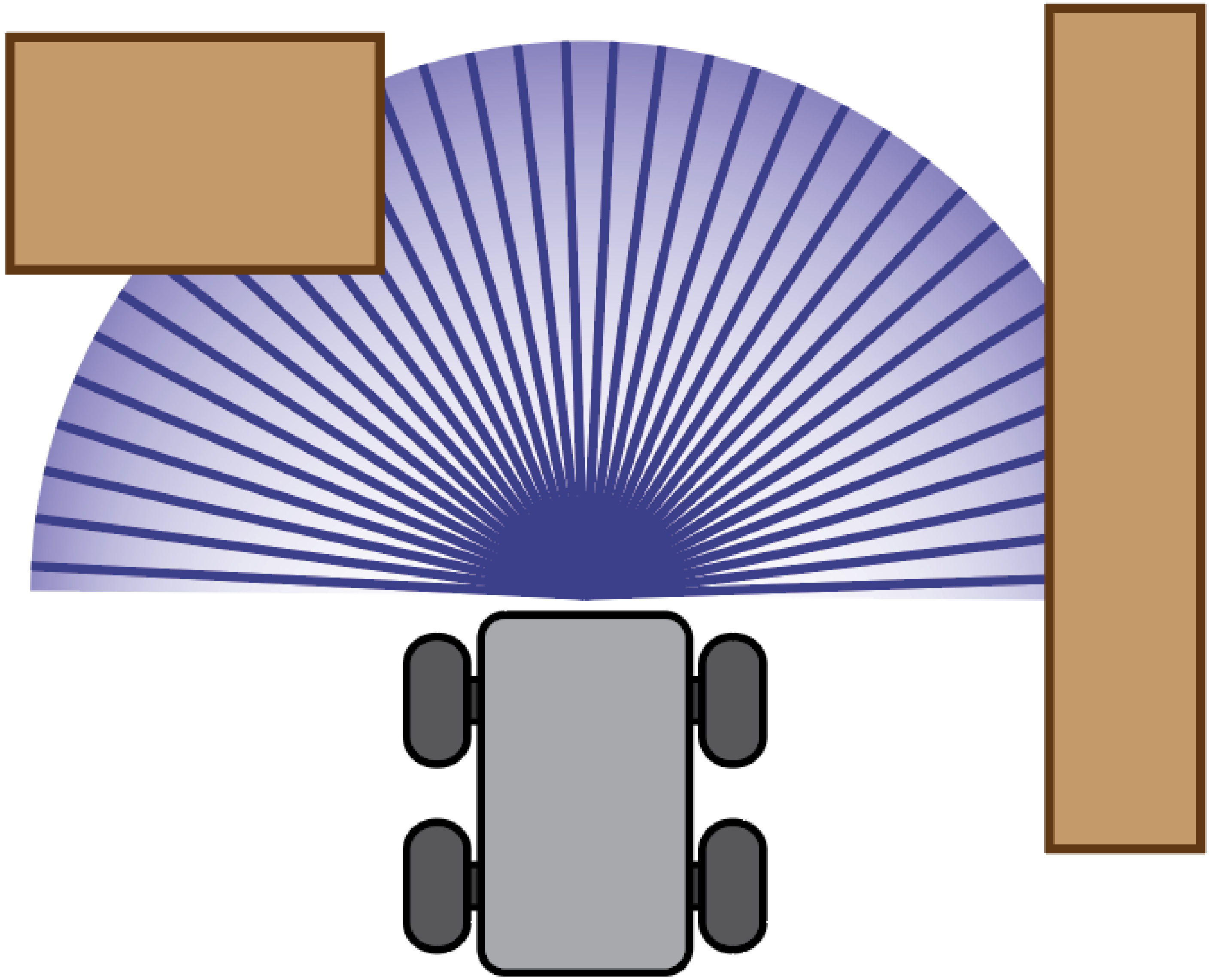
For instance, if the max distance is 10, then we can get the ranges by:

```
range_R  = min(msg.ranges[0],10)
range_FR = min(msg.ranges[1],10)
range_F  = min(msg.ranges[2],10)
range_FL = min(msg.ranges[3],10)
range_L  = min(msg.ranges[4],10)
```



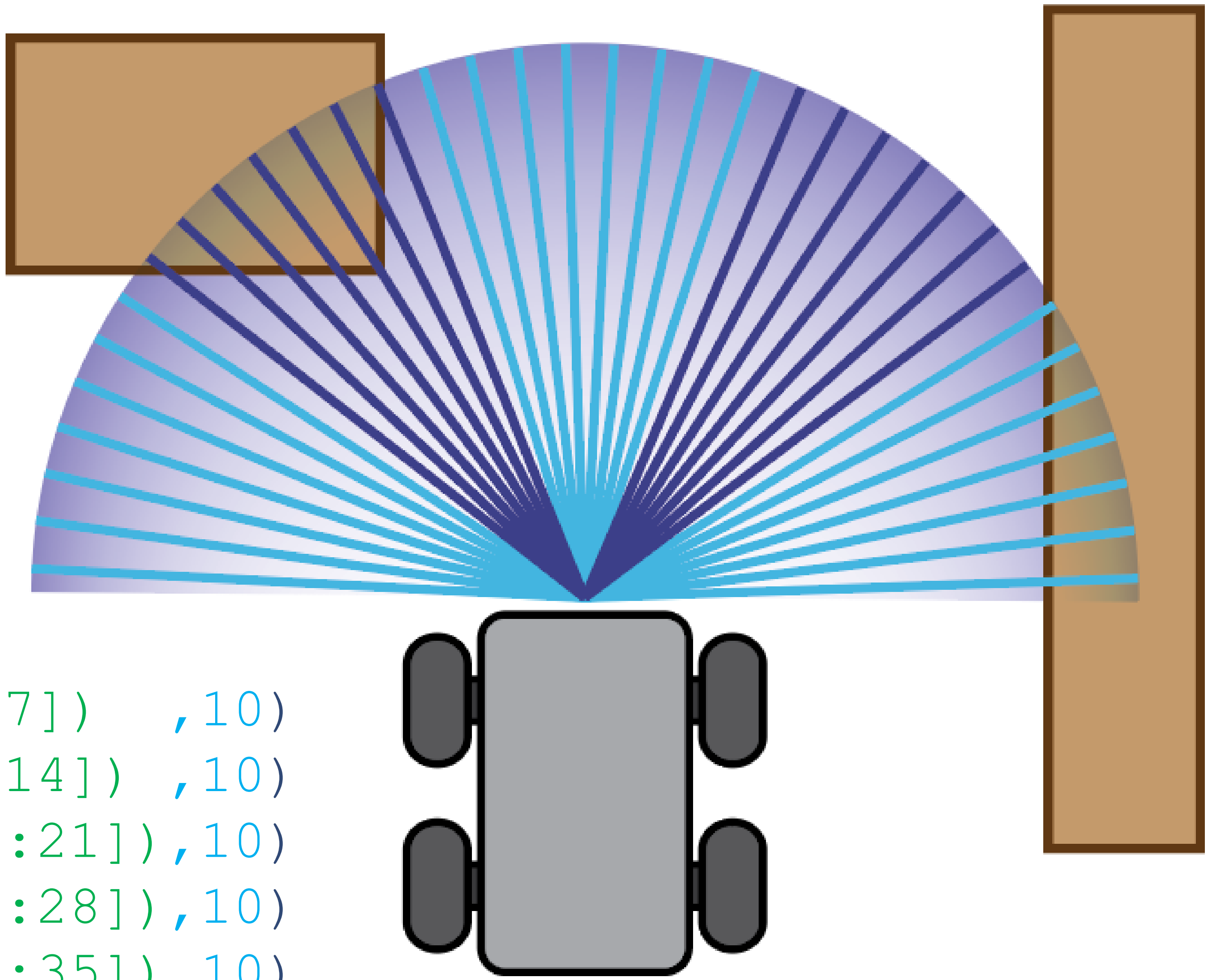
Laser Scan in ROS

In practical situations, we use a large number of laser samples for greater effectiveness and for implementing more complex algorithms



Laser Scan in ROS

The laser samples are divided into **regions**. We can get *region values* by using the minimum value of the laser sample in that region



For instance, to get region values for 36 laser samples:

```
R  = min(min(msg.ranges[0:7])  ,10)
FR = min(min(msg.ranges[7:14]) ,10)
F  = min(min(msg.ranges[14:21]),10)
FL = min(min(msg.ranges[21:28]),10)
L  = min(min(msg.ranges[28:35]),10)
```

Laser Scan in ROS

```
R  = min(min(msg.ranges[0:7])   ,10)
FR = min(min(msg.ranges[7:14])  ,10)
F  = min(min(msg.ranges[14:21]),10)
FL = min(min(msg.ranges[21:28]),10)
L  = min(min(msg.ranges[28:35]),10)
```

We can generalize the above 5 regions for x number of laser samples:

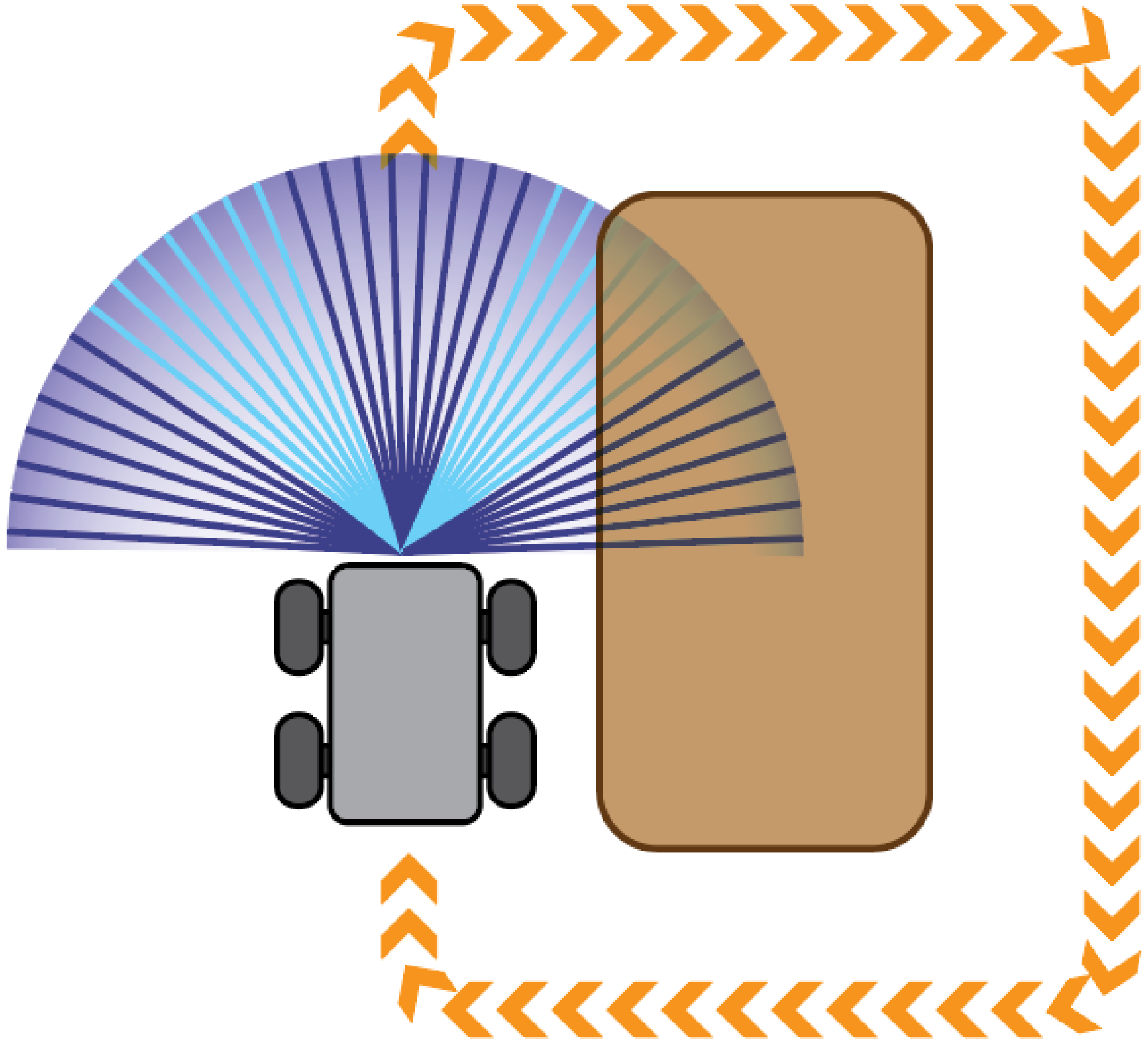
```
x  = len(msg.ranges)
R  = min(min(msg.ranges[0*x//5 : (1*x-1)//5]), max_dist)
FR = min(min(msg.ranges[1*x//5 : (2*x-1)//5]), max_dist)
F  = min(min(msg.ranges[2*x//5 : (3*x-1)//5]), max_dist)
FL = min(min(msg.ranges[3*x//5 : (4*x-1)//5]), max_dist)
L  = min(min(msg.ranges[4*x//5 : (5*x-1)//5]), max_dist)
```

Wall Following

In this lab, we will look at wall following around obstacles which can be either convex or concave in shape

A shape is called convex if every straight line made by any two points in the shape, will pass through the shape only once

If the line passes through the object more than once, the shape is concave



Lab Tasks

- Download the manual from LMS
- Perform the Lab Tasks as given in the manual and submit it on LMS
- Remember to execute scripts with the terminal