# Lab 4

# Inertial Measurement Units

This lecture is part of the RACECAR-MN introductory robotics course.
You can visit the course webpage at mitll-racecar-mn.readthedocs.io.

# Objectives

**Main Objective**: Use the RACECAR's IMU data to prevent rolling

## Learning Objectives

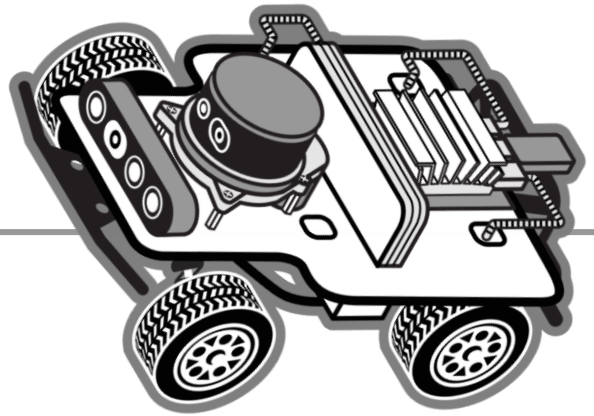
- Use the Physics module to retrieve linear acceleration and angular velocity

# Roll Prevention

- Vehicle rollover is a threat for cars and trucks
  - Turning too quickly can cause a vehicle with a high center of mass to roll

# Roll Prevention

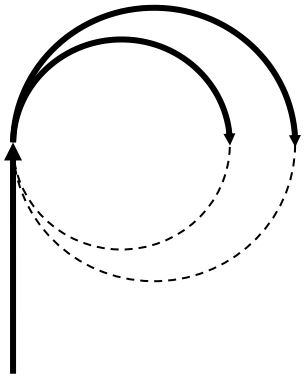
- It is difficult to roll the RACECAR-MN due to its low center of mass, but we will model this problem by artificially raising the center of mass in RacecarSim



# Rolling Prevention

- Rolling is caused by **inertia**
  - When a car turns the inertial force continues linearly, which creates an outward force
- There are various ways to prevent rolling:

Super elevation



Wider turning

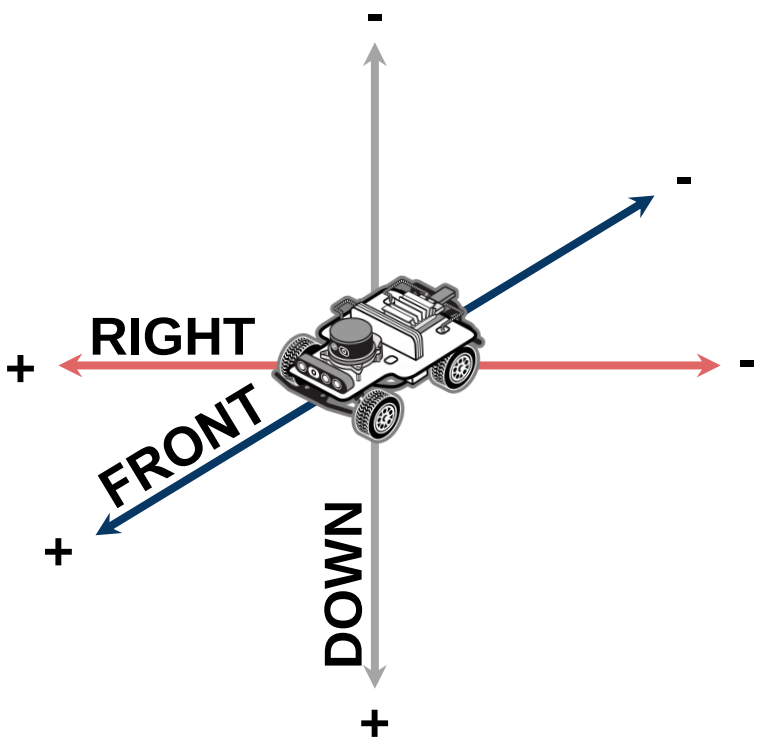
Reducing speed

# Inertial Measurement Unit (IMU)

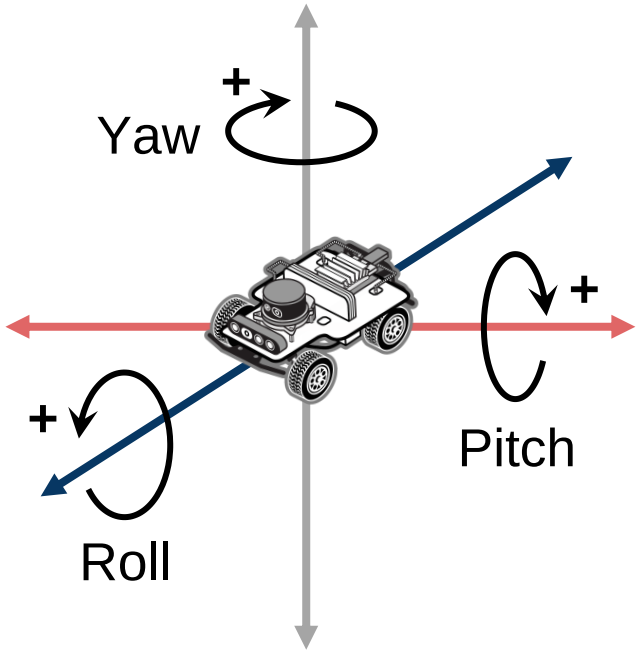
- Uses accelerometers and gyroscopes to measure:
  - **Linear acceleration** (acceleration data)
  - **Angular velocity** (gyro data)

# Inertial Measurement Unit (IMU)

## Accel

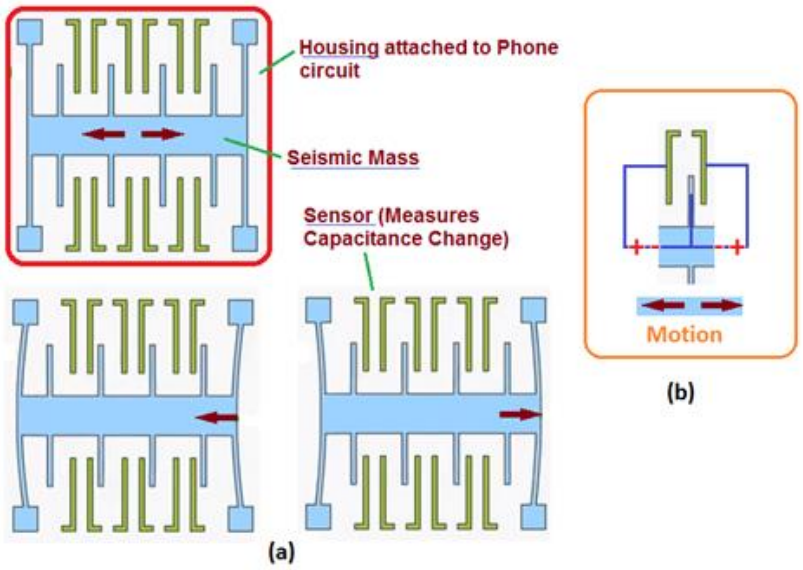



## Gyro

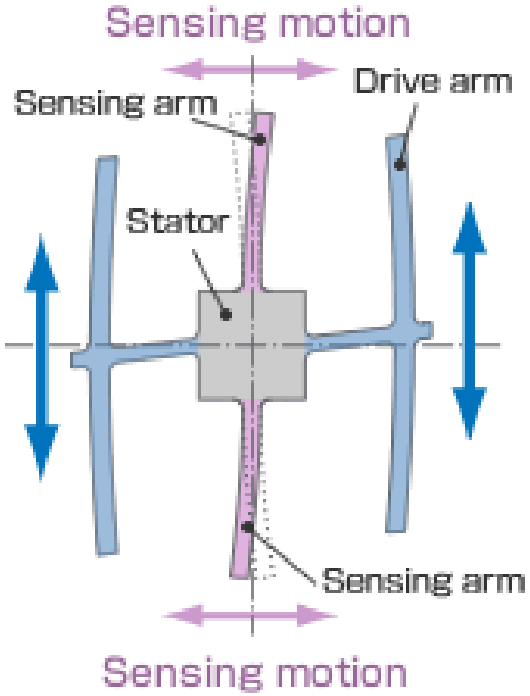


# Inertial Measurement Unit (IMU)

## Accelerometer




## Gyroscope



# Using IMU Data

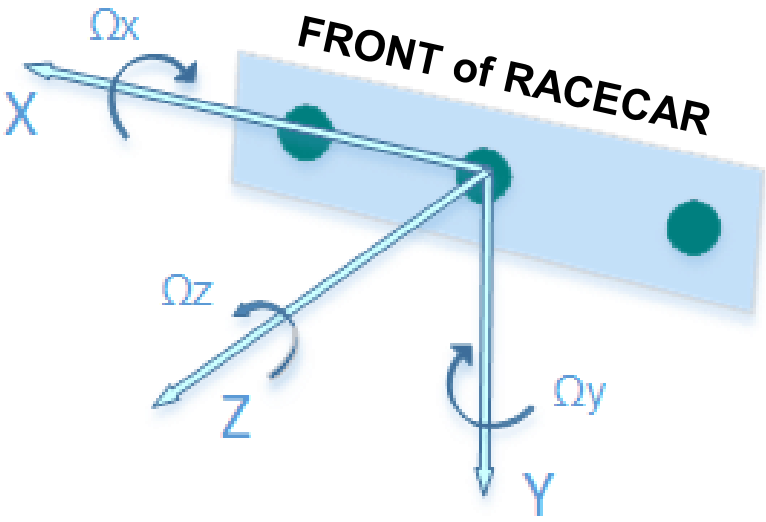
Group activity

- What benefits are there to having linear acceleration?
- What benefits are there to having angular velocity?
- What concerns do you have about using the IMU data?

# Intel RealSense D435i

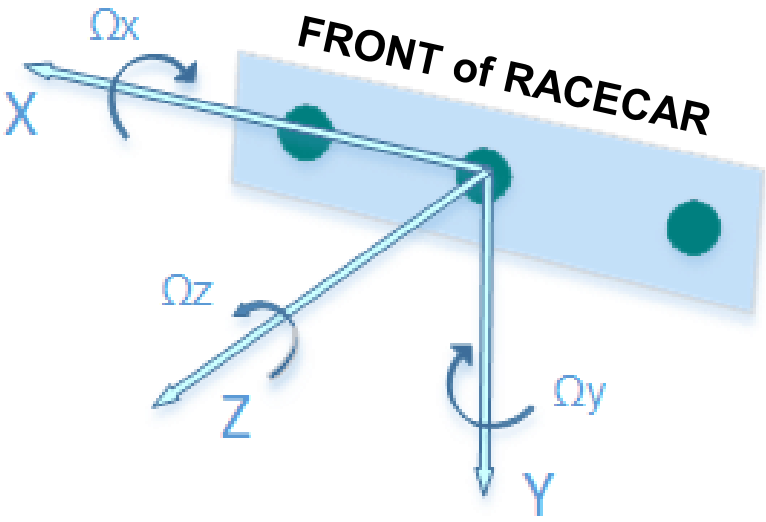
# Acceleration

- Has three values (in $m/s^2$):
  - Linear acceleration on X-axis, which points to the **right**
  - Linear acceleration on Y-axis, which points **down**
  - Linear acceleration on Z-axis, which points **forward**



# Gyro

- Has three values (in radians/sec) :
  - Angular velocity about X-axis (denoted by $\Omega x$)
  - Angular velocity about Y-axis (denoted by $\Omega y$)
  - Angular velocity about Z-axis (denoted by $\Omega z$)



# Physics Module

- Retrieves IMU data
- Public Interface
  - `get_linear_acceleration()`
  - `get_angular_velocity()`

# Examples

Group activity

```
# Example 1
def update():
    accel = rc.physics.get_linear_acceleration()
    ang_vel = rc.physics.get_angular_velocity()

    if accel[2] > 0.10:
        print("Kachow!")

    if ang_vel[0] > 0.25:
        rc.drive.stop()
```

# Examples

Group activity

```
foo = 0

def update():
    global foo

    ang_vel = rc.physics.get_angular_velocity()
    foo += ang_vel[1] * rc.get_delta_time()

    if foo < math.pi / 2:
        rc.drive.set_speed_angle(1, 1)
    else:
        rc.drive.set_speed_angle(1, 0)
```