WORCESTER POLYTECHNIC INSTITUTE

LEHR UND KUNST

1865

# Teleoperation of A Robotic Arm Using IMU

**Team 7**

Arpit Gupta Daniel Wivagg Nalin Raut Nishant Shah Sanjuksha Nirgude

# Introduction

## Why is it important to solve this problem?

Human safety has always been an issue in hazardous environmental working in any field. To overcome these problems of human safety robot technology can be proved to be a boon.

## **Approach:**

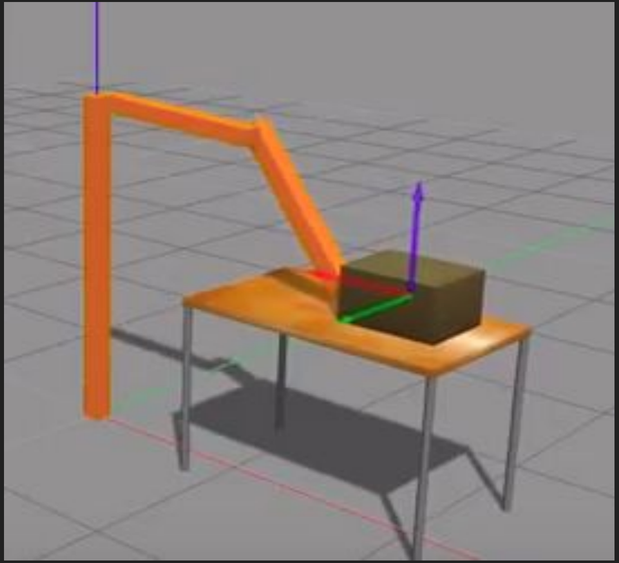
Teleoperation can be a very good way to operate robots from a distance and at the same time keeping the human operator in a safe distant environment. The most basic way to do this is using an IMU Inertial Measurement Unit, which gives us 9 degrees of freedom. So the flexibility of this particular sensor enables us to operate a robot easily.

# Problem Statement

This task involves a robot in simulation, which is a 4 degree of freedom arm that is operated by the IMU sensor using an arduino board as our controller.

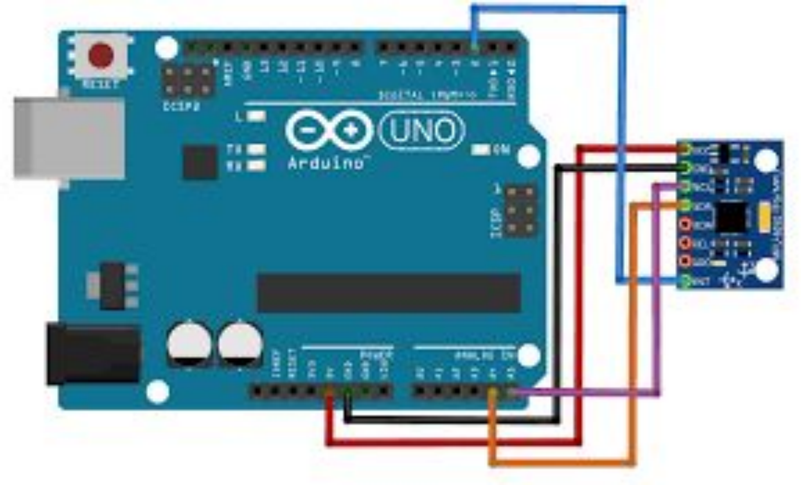
This robot is known as the rrbot. We use the gazebo simulator for our simulation and also the Robot Operating System (ROS) for the communication between the IMU sensor and the rrbot.

The robot uses the IMU signals to move its arm and push an object of the table.



Rrbot simulation Setup

# Hardware Setup



1. **Arduino UNO R3**
2. **IMU sensor (Accelerometer, Gyroscope, Compass)**
3. **Push Button**
4. **10 k Ω resistor**

# Software Setup

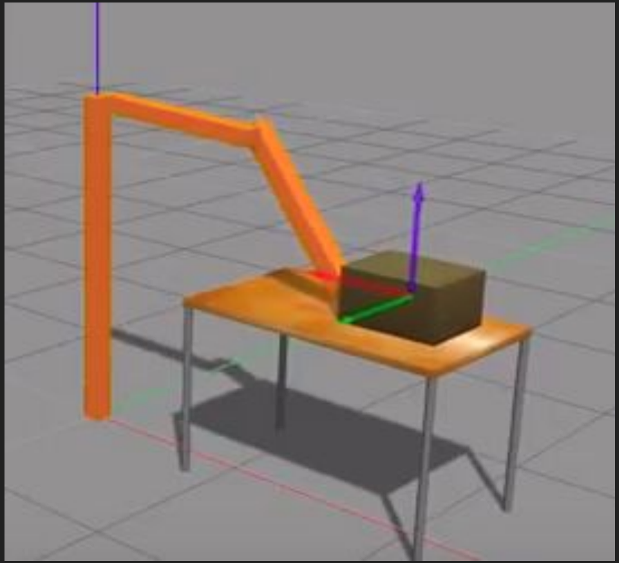
**Software:**

**1. Gazebo Simulation Environment.**

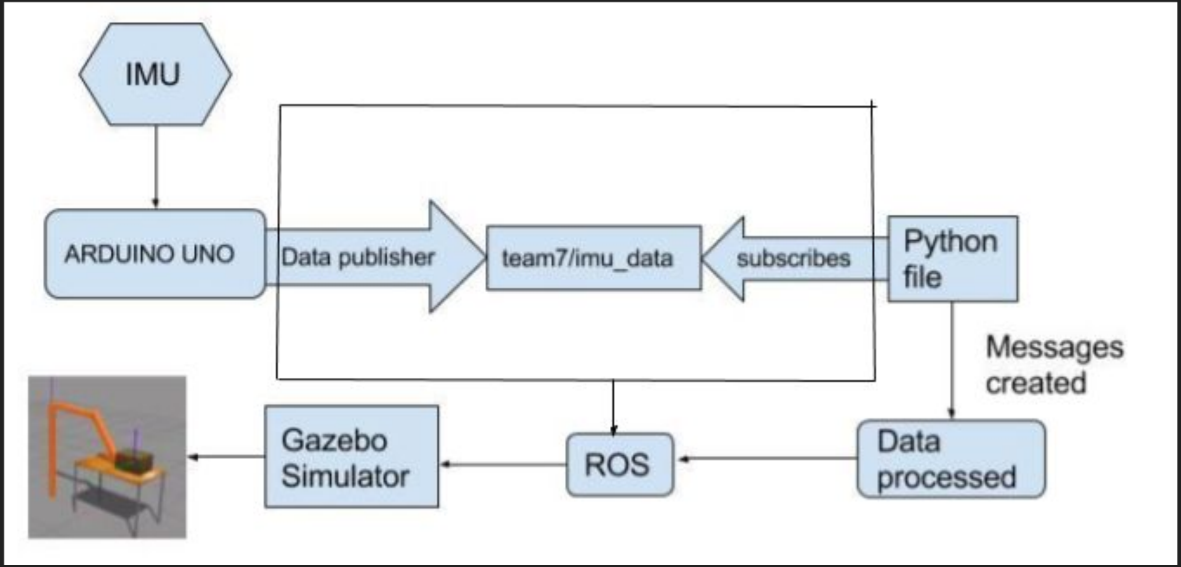
**2. Arduino IDE.**

**3. Robot Operating System (Kinetic and Indigo ) on Ubuntu 16.04 and 14.04 resp.**

# Demonstration



# Architecture



# Problems encountered and workarounds

-Arduino constantly loses connection over Rosserial (probably because memory was low)
-The sketch was originally too large for the Arduino’s memory so some adjustments had to be made to shrink it
-We had trouble finding a good package for a robot model that we could operate. It needed to have good message types that we could access and run on the computers we had.

**Solutions**:
- We stored the data in a ROS bag to avoid the connection problem. This allowed us to play back a constant stream of data.
-The data types in the sketch and messages used just had to be adjusted to conserve memory space
-We found a few robots that met our criteria and settled on the best one, although we have implementations for all 3.

# Future Work

- Using kalman filter for smooth motion of the arm
- Using a gripper as an end effector to pick and place the objects
- Use more than one IMU sensors to capture more degree of freedoms from the operator’s arm
- Mapping of sensor data with the robot joints
- Create a simple robot arm to learn more about ROS URDF and package creation

# THANK YOU!