EN4594 Autonomous Systems
Laboratory Sheet
Practical No. 1

Title: Zumo ROS Interface and Calibration of IMU Sensors

1. Introduction

The Zumo 32U4 robot is a complete, versatile robot controlled by an Arduino-compatible ATmega32U4 microcontroller. At the heart of the Zumo 32U4 is an integrated ATmega32U4 AVR microcontroller from Atmel, along with dual H-bridge drivers that power the robot's motors. The robot also features a variety of sensors, including quadrature encoders and inertial sensors on the main board, along with reflectance and proximity sensors on the front sensor array.

The Zumo 32U4 includes the following on-board inertial sensors that can be used to determine its orientation: ST LSM303D chip combines an accelerometer sensor and a magnetometer sensor, and ST L3GD20H chip is a gyroscope sensor. Nevertheless, the sensors need to be calibrated before used in any application.

The process of checking the accuracy of a measuring instrument, by comparison with a standard input is known as calibration. Moreover, calibration also deals with adjustment of the instrument/sensor to bring it into alignment with the standard input.

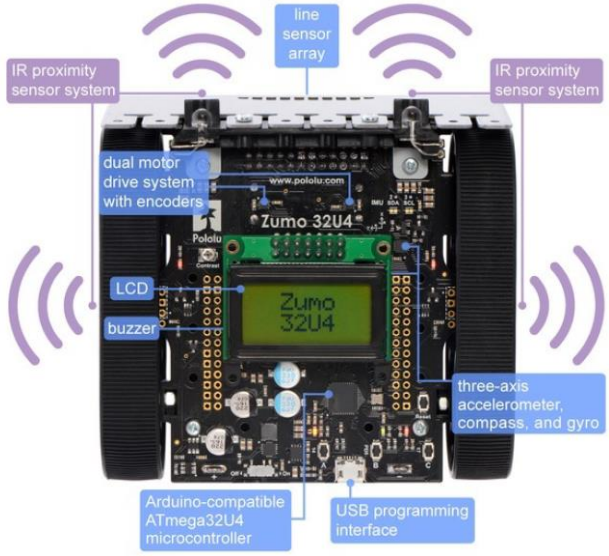

Figure. 1: Pololu Zumo 32U4 Robot

In this practical you will learn the following:

- Connection to Zumo 32U4 robot via a ROS interface
- Calibration of IMU sensors

2. Theoretical Background

This section describes the ROS interface and the related theoretical background needed to complete this practical.

2.1. ROS Interface

As depicted in Fig. 2, multiple nodes communicate with each other to achieve required functionality. `zumo_serial_node` is responsible for querying Zumo sensors and publishing their data as ROS messages. It can also accept motion commands to locomote the Zumo robot. In the example given in Fig. 2, zumo_acc_calib node performs IMU accelerometer calibration.

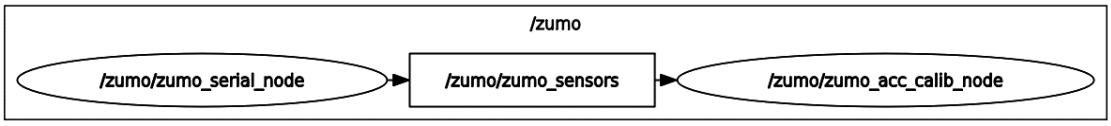

Figure 2: ROS node graph

2.2. Calibration

Calibration refers to the process of comparing measurements or observations of a device or system to known standards or reference values, and making adjustments to ensure accuracy and reliability. This adjustment is done by establishing a relationship between the output of the instrument to a known input. Once such a relationship is established, we are able to correct sensor readings to reflect the correct measurement.
We will perform two types of calibration for the IMU sensor: accelerometer calibration and magnetometer calibration.

- Accelerometer calibration
  To calibrate the accelerometer, we make use of known gravitational acceleration (direction and magnitude): 9.81ms^{-2} $\downarrow$

$$\begin{bmatrix} a_x^{obj} \\ a_y^{obj} \\ a_z^{obj} \end{bmatrix} = R_s^{obj} \cdot \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \gamma \end{bmatrix} \cdot \begin{bmatrix} a_x^s - a_{x-os} \\ a_y^s - a_{y-os} \\ a_z^s - a_{z-os} \end{bmatrix} \quad (1)$$

where obj: object, s: sensor, and os: offset.

R_s^{obj} rotation matrix accounts for the misalignment. The next matrix that includes α, β, γ performs required scaling. Finally, the offsets are added. Equation (1) can be simplified to the one given in eq. (2).

$$\begin{bmatrix} a_x^{obj} \\ a_y^{obj} \\ a_z^{obj} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \end{bmatrix} \begin{bmatrix} a_x^s \\ a_y^s \\ a_z^s \\ 1 \end{bmatrix} = C \begin{bmatrix} a_x^s \\ a_y^s \\ a_z^s \\ 1 \end{bmatrix} \quad (2)$$

where we formulate the C calibration matrix.

We use the least square method to find the calibration matrix in eq. (2), as explained in detail in the *Least Squares* lecture.

- Magnetometer calibration
  We use a simple technique to calibrate the IMU's magnetometer. Among different types of distortion sources, we focus on removing *hard iron* distortion. Hard iron distortion is caused by the materials which emit a magnetic field, such as magnets, speakers, or motors, nearby. This distortion causes an offset in the readings (Fig. 3), and is easily handled by calculating the offset and subtracting its value from the sensor output.

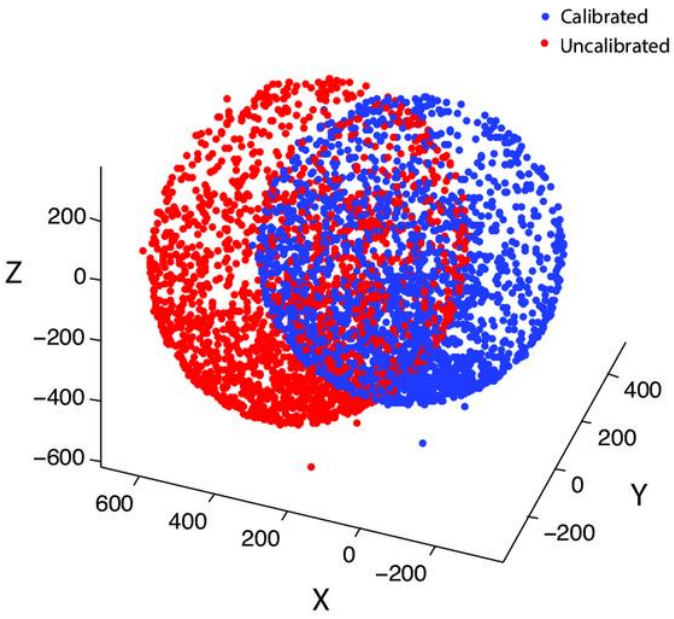

Figure 3: Calibrating the IMU magnetometer

3. Procedure

This section describes the necessary procedure for IMU calibration.

3.1. Folders

Copy the following folders into your ROS workspace `src` folder.

```
zumo ------- zumo_calibration
        |---- zumo_defs
        |---- zumo_launch
        |---- zumo_msgs
        |---- zumo_serial
```

3.2. Install Other Required Libraries

- Serial Communication
  Make sure to have libserial library installed for serial communication. Open a new terminal and type:
  `sudo apt-get install libserial-dev`

- `xterm` for launch
  `xterm` will help us receive keyboard input from terminal when launching nodes.
  `sudo apt-get -y install xterm`

3.3. Rules for Device Manager

These rules are used to define how devices are identified and managed by the Linux system. Copy the provided `a-star.rules` file into Ubuntu `udev` rules folder:
e.g.

```
sudo cp -a ./a-star.rules /etc/udev/rules.d/
```

3.4. Build the Nodes

Open a new terminal and navigate to your ROS workspace.
e.g. `cd ~/ros2_ws/`
To build the workspace use the following command:

```
colcon build --symlink-install
```

Remember to use `--symlink-install` parameter to create symbolic links (symlinks) instead of copying files from the build directory to the install directory. These symlinks point to the corresponding files in the build directory. As a result, any changes made to the built artifacts in the build directory are immediately reflected in the installed artifacts, as they are essentially the same files. This will help you make changes to launch files with immediate effect without having to build the workspace, again.
If you encounter any compilation errors, contact the instructor to resolve them.

3.5. Zumo Connection

- Connect the Zumo 32U4 robot to the computer using the USB A to Micro-B cable.
  If you are using Oracle VM VirtualBox, you need to establish the connection manually as shown in Fig. 4, where you will choose Arduino LLC Arduino Leonardo [0100].

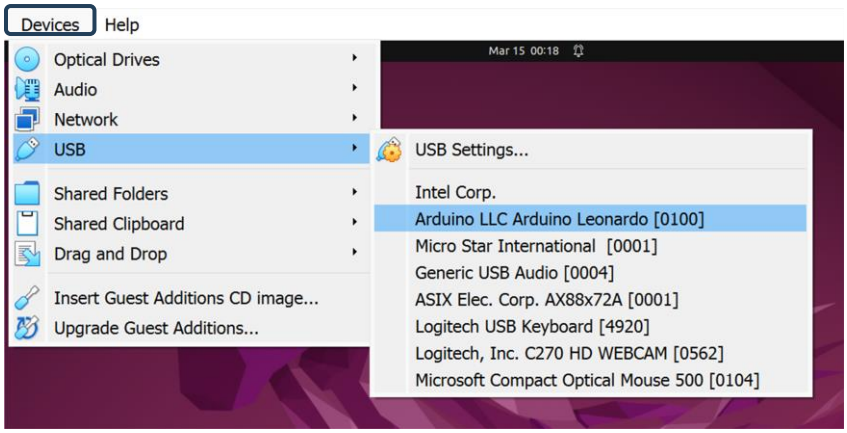

Figure 4: Zumo Connection in Oracle VM VirtualBox

- To check the USB connection, open a new terminal and type:
  `cd /dev`
  `ls`
  Check whether `ttyACM0` (`ttyACM1`, `ttyUSB0` or something similar) exists after connecting the Zumo robot.
- Give permission to access the device.
  `sudo chmod a+rw /dev/ttyACM0`

3.6. Running ROS Nodes

- Follow the instructions given in section 4.1 to start the Zumo ROS interface.
- Follow the instructions given in section 4.2 to observe Zumo sensor data.
- Perform sensor calibration for i) accelerometer ii) magnetometer as explained in section 4.3. Obtain the calibration parameters.

4. ROS Reference

This section describes ROS commands to startup ROS nodes and perform IMU calibration.

4.1. Zumo Interface

- Open a new terminal and type:
  `ros2 launch zumo_launch zumo_serial_node.launch`
  This command will establish serial communication with the ZUMO robot.

  Note: If the serial port is different from `/dev/ttyACM0`, you need to change `zumo_serial_port` parameter in `zumo_serial_node.launch` file in `zumo_launch` package. You may use IDE opened in the ROS workspace.
  Open up a new terminal and type:
  `cd ros2_ws/`
  `code`

4.2. Observing Zumo Sensor data

- Run the Zumo interface as described in section 4.1.
- Open up a new terminal and type:
  `ros2 topic echo /zumo/zumo_sensors`
  This command will echo sensor data in the terminal. You may rotate the track wheels or change the orientation of the robot to observe changes in sensor data.
- Open up a new terminal and type:
  `rqt`
  This command will start up the ROS GUI for data plotting. Select Plugins→Visualization→Plot. Choose any topic you want to visualize from the below list.

  Topics

  acceleration: `/zumo/zumo_sensors/ax`
  `/zumo/zumo_sensors/ay`
  `/zumo/zumo_sensors/az`

  gyro: `/zumo/zumo_sensors/gx`
  `/zumo/zumo_sensors/gy`
  `/zumo/zumo_sensors/gz`

  magnetic: `/zumo/zumo_sensors/mx`
  `/zumo/zumo_sensors/my`
  `/zumo/zumo_sensors/mz`

  encoder: `/zumo/zumo_sensors/enc_left`
  `/zumo/zumo_sensors/enc_right`

  Note: To stop any node, press `Ctrl+c` on the terminal.

4.3. Running Accelerometer and Magnetometer Calibration Nodes

- Run the Zumo interface as described in section 4.1.
- As explained in section 4.2, keep open a new terminal with `/zumo/zumo_sensors` topic echo to make sure that sensor data from Zumo is arriving continuously without interruption.
- Open up a new terminal and type:
  `ros2 launch zumo_launch zumo_acc_calib.launch`
  This command will start the accelerometer calibration node. Follow the instructions that appear on the screen. Once the calibration is completed, save the elements of the calibration matrix for future use.

  Note
  If `xterm` does not startup correctly (especially in Oracle VM VirtualBox) you can just run the node as follows:
  `ros2 run zumo_calibration zumo_acc_calib_node`

- Open up a new terminal and type:
  `ros2 launch zumo_launch zumo_mag_calib.launch`
  This command will start the magnetometer calibration node. Rotate the robot in all 3 axes and obtain the minimum and maximum values for all three axes. Once the calibration is completed, save all the minimum and maximum values for future use.

  Note
  To stop any node, press `Ctrl+c` on the terminal.