**Laboratory 2 - Open Loop Control**

2.1 Introduction

In this lab we are going to implement an open loop stabilizer using the IMU and servo motors.

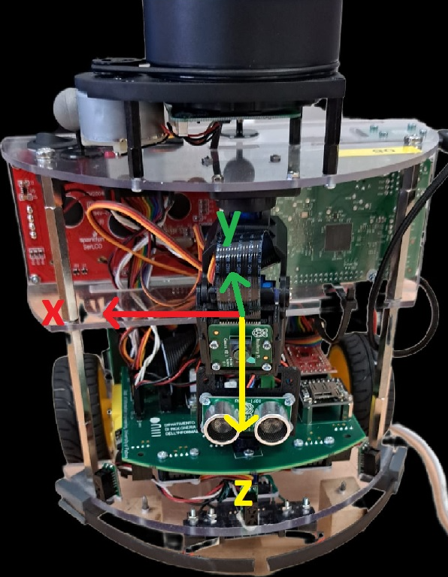
**IMU.**

IMU stands for Inertial Measurement Unit. It is a device that is used to measure and report an object's specific force, angular rate, and sometimes the magnetic field surrounding it. IMUs are commonly used in various applications, including robotics, navigation systems, virtual reality, augmented reality, and motion capture.

An IMU typically consists of several sensors that work together to provide information about an object's motion and orientation:

1. Accelerometer: Measures linear acceleration along different axes (typically three axes: X, Y, and Z). It detects changes in velocity and orientation, allowing the IMU to determine the object's acceleration and tilt.
2. Gyroscope: Measures angular velocity or rate of rotation around different axes. It provides information about the object's rotational motion and helps track changes in orientation.
3. Magnetometer: Measures the strength and direction of the magnetic field around the object. It is often included in IMUs to provide additional information for orientation estimation, especially when used in combination with accelerometers and gyroscopes.

By combining the data from these sensors, an IMU can estimate the object's orientation, position, and velocity relative to its initial state.



The axis of the accelerometer

• Tilt refers to the vertical movement of the camera, where the camera is moved up or down.

• Pan refers to the horizontal movement of the camera, where the camera is rotated around its vertical axis to the left or right.

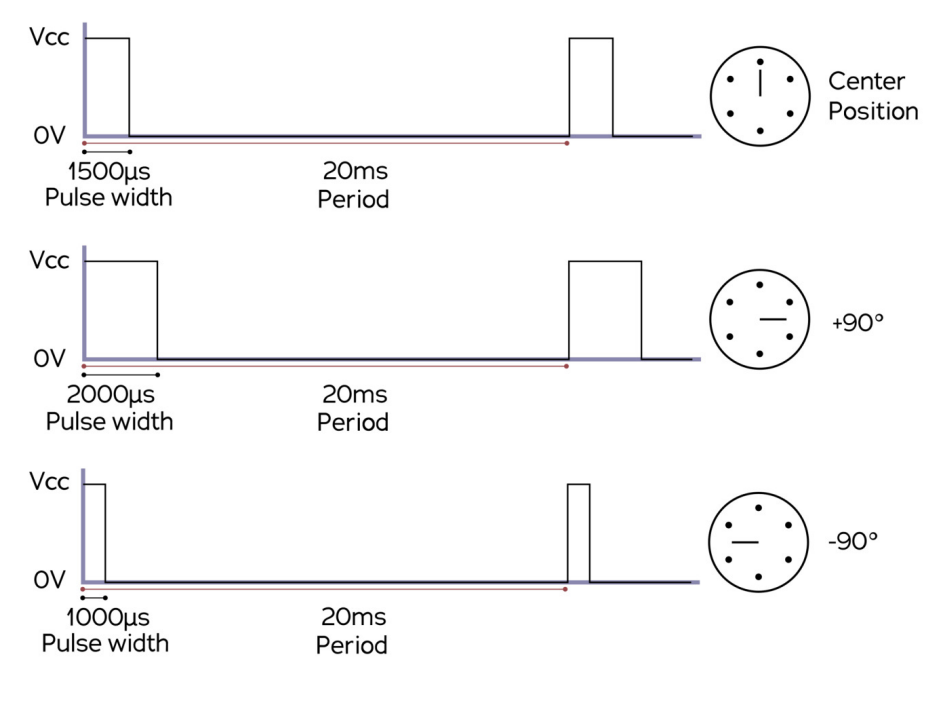
**Servo Motor**

A servo motor is a type of motor that is widely used in various applications, particularly in robotics, automation, and control systems. It is designed to provide precise control over angular or rotational movement.

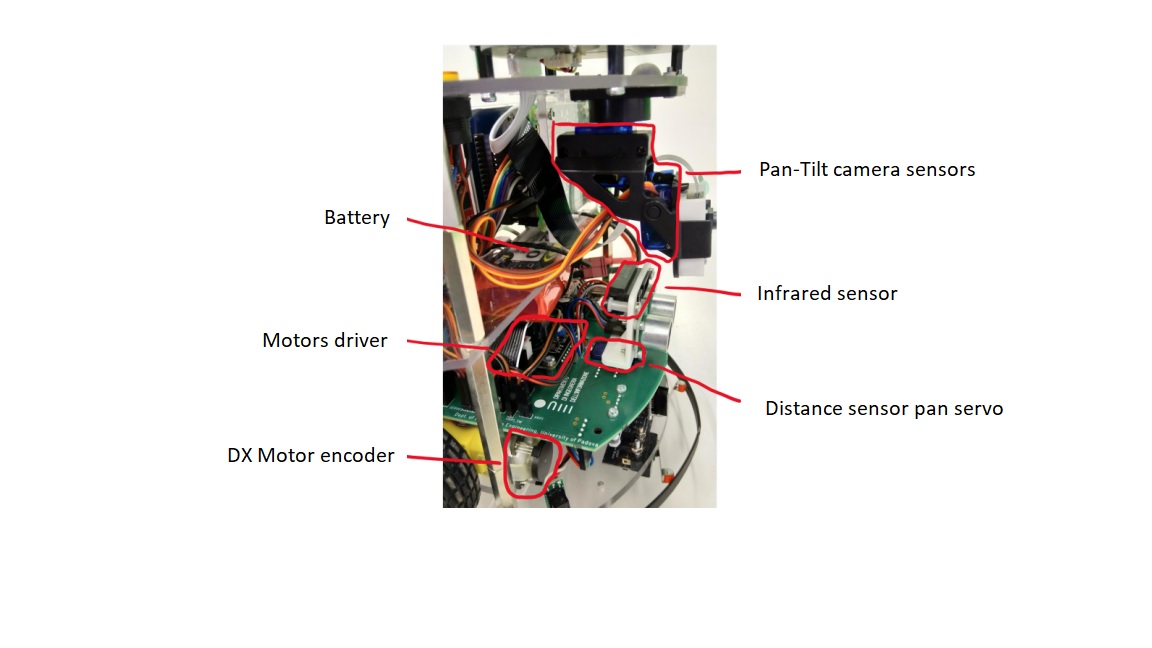
The distinguishing feature of a servo motor is its ability to maintain a specific position or follow a desired trajectory with great accuracy. It achieves this through a closed-loop control system, which continuously compares the actual position of the motor shaft with the desired position and adjusts the motor's output accordingly.

Here are some key components and characteristics of servo motors:

1. DC Motor: Most servo motors are based on a DC motor as the primary driving mechanism. The DC motor converts electrical energy into rotational motion.
2. Gear Train: Servo motors often include a gear train that reduces the motor's high-speed, low-torque output to a lower speed with higher torque. This gearing mechanism enables the servo motor to generate more precise and controlled movements.
3. Position Feedback Sensor: A servo motor typically incorporates a position feedback sensor, such as an encoder or a potentiometer. This sensor provides information about the current position and velocity of the motor shaft. The feedback data is used by the control system to determine if the motor needs to adjust its position.
4. Control Circuitry: The control circuitry of a servo motor includes a microcontroller or a dedicated servo controller. It receives the control signal or command from an external device, such as a microcontroller or a computer, and generates the appropriate electrical signals to drive the motor.
5. Pulse Width Modulation (PWM) Signal: Servo motors commonly utilize a PWM signal to control their position and speed. The control signal consists of a series of pulses with varying widths, where the width of each pulse determines the desired position. The control circuitry interprets the pulse width and adjusts the motor's position accordingly.



On the TurtleBot these are two servo that control the tilt and pan of a camera as shown in the figure below.



**Exercises**

Exercise 1. Develop a control law to stabilize the camera’s tilt by utilizing data from an IMU.

First of all, we have to keep the camera aligned (0 degree) with the horizon. For this purpose, we should obtain the accelerometer and gyroscope data from the robot.

int8\_t tilt = 0;

**float** angle = 0;

**while** (1)

{

HAL\_Delay(20);

bno055\_convert\_double\_accel\_xyz\_msq(&d\_accel\_xyz);

angle = (**asin**(d\_accel\_xyz.y/ 9.81)) \* 180 / 3.14;

tilt = -angle;

logger\_data.u1 = tilt;

logger\_data.u2 = d\_accel\_xyz.y;

ertc\_dlog\_send(&logger, &logger\_data, **sizeof**(logger\_data));

ertc\_dlog\_update(&logger);

\_\_HAL\_TIM\_SET\_COMPARE(&htim1,TIM\_CHANNEL\_3,(uint32\_t)saturate((150+tilt\*(50.0/5.0)), SERVO\_MIN\_VALUE, SERVO\_MAX\_VALUE));

}

As you see, first we defined tilt and angle globally. To obtain the accelerometer data we use “bno055\_convert\_double\_accel\_xyz\_msq”and “d\_accel\_xyz” as a pointer to a structure as input.

The formula for finding the tilt of Tbot is:

Which is the tilt angle, is the acceleration measured on the y axis and g is gravity acceleration (=9.81). To convert the from radiant the equation should be multiplied by .

As we want to put the camera aligned we should put the tilt equal to, otherwise the camera turns reverse.

For moving the camera we use “\_\_HAL\_TIM\_SET\_COMPARE” Which is a HAL library macro used for setting the compare value of a specific channel of a hardware timer on a microcontroller.

To record the data, we use datalogger. As all the configuration for using the datalogger is already done, first we define a structure to send the data:

**struct** ertc\_dlog logger;

**struct** datalog

{

**float** u1, u2 ;

} logger\_data;

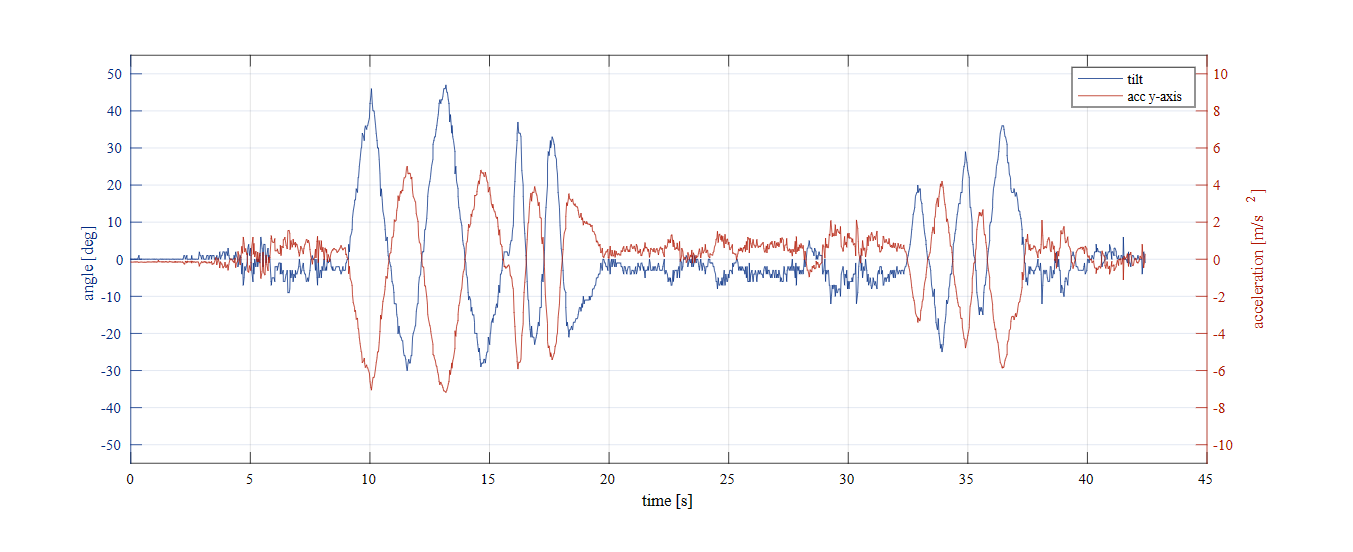
Then we use logger\_data.u1 as the tilt data and logger\_data.u2 as the acceleration measured on the y axis data in the while loop.

To plot the data we use Matlab and the serial datalogger is invoked using serial\_datalog() function as below :

data = serial\_datalog('COM9',{'2\*single','2\*single'}, 'baudrate', 115200)

“**COM9**” is the port of the serial datalogger. **“single,single”** specifies the data format to be logged. It indicates that two data values will be logged in each iteration of the logging process. In this case, the logged data is expected to be of type single (a single-precision floating-point number).

**“baudrate”, “115200”** sets the baud rate of the serial communication. The baud rate determines the speed at which data is transferred over the serial port. In this case, the baud rate is set to 115200 bits per second.



**Bonus**

Bunus1. Implement a control algorithm that implement a “smooth pan” control (i.e. a control system that compensate sharp horizontal rotations of the TBot) using data from the gyro.

Process is like exercise 1 :

int8\_t pan = 0;

**float** angle = 0;

**while** (1)

{

HAL\_Delay(20);

bno055\_convert\_double\_gyro\_xyz\_rps(&d\_gyro\_xyz);

angle = d\_gyro\_xyz.z \* 180 / 3.14;

pan = -angle/8;

logger\_data.u1 = pan;

logger\_data.u2 = d\_gyro\_xyz.z;

ertc\_dlog\_send(&logger, &logger\_data, **sizeof**(logger\_data));

ertc\_dlog\_update(&logger);

\_\_HAL\_TIM\_SET\_COMPARE(&htim1,TIM\_CHANNEL\_3,(uint32\_t)saturate((150+pan\*(50.0/55.0)), SERVO\_MIN\_VALUE, SERVO\_MAX\_VALUE));

}

As it is clear to obtain the we have to put the gyroscope data measured in z axis and we put the pan equal to which the negative sign is because of turning up and down properly and is divided to 8 which is the proper gain.

And inside the “\_\_HAL\_TIM\_SET\_COMPARE()”, as the channel 3 is controlling the pan, we should put “TIM\_CHANNEL\_3” instead of “TIM\_CHANNEL\_2” which is controlling the tilt.

