

Communication and control algorithm :

Hardware and connections : We will be having a raspberry pi and arduino . The raspberry pi will be connected to the arduino board through vin pin which will power the arduino(for this time) and also establish serial communication between arduino and raspberry pi.

We will be flashing raspberry pi with ubuntu 20.04 LTS. We will connect lidar and power supply to the raspberry pi board. The arduino board will be having a motor controller which will control 4 DC motors. We will be having encoders on all 4 motors which are connected to arduino board.

Coding Part:

Rpi side: From rpi side we will be having a code that is basically a ros code that will be subscribing, publishing and establishing serial communication. In the code we have established serial communication using line " ser = serial.Serial("/dev/ttyUSB0", 9600)"

Here "/dev/ttyUSB0" is a port which is used for connection between USB and Serial adapter with baud rate of 9600. We are having a function "received info()" which is used to receive data from arduino. The communication happens in a way "<var1,var2,var3,var4>" where var* is data to be transferred and it is separated by ",", also starting point will be "<" and endpoint will be ">" so that we can differentiate between the different inputs. Then we have "wait_for_arduino()" function which will be waiting for the message from the arduino. We have a shut_down function which will be closing the serial communication.

We have a main_fun() which declares subscribers and publishers . Also this function is responsible to send the data from rpi side to arduino in the above mentioned format. Also this function processes the data which we got from the arduino(encoder data) and publishes .

Arduino side : In arduino side we have a code that takes input from rpi and processes the input and current data , measures the error and feeds it into PID and controls the output. Here initially we are having declarations of arduino pins . Next we are having motor class which contains setDesiredSpeed() which takes care that the output is within the range of maximum speed and minimum speed, also the class motor has computeOutput() function which will be containing the PID code. Then we have setup function for setting up the pins and initiating serial communication.

We have two other functions one is recvWithStartEndMarkers() and other is parseData() . First one is responsible for collecting the data that is sent over serial communication and it stores the data in receivedChars array. Second function splits the data and separates individual velocity inputs.

In the loop() function we are giving input to the motor controller with the data that we got till now and also send the data to arduino. Here the code is incomplete because it varies depending on the motor controller, also the code for encoder varies depending upon the encoder . We need to add the attachInterrupt events in PID code to count for the encoder values.