ITSP Final Documentation

NeuroClone

Team Inception

ITS21009

Keywords

EEG, Open Manipulator, ROS, CNN, Deep Learning, PyTorch, Paralyzed, Robot, Gazebo, Python

Team Member Name	Roll Number	Email-Id
Om Mihani	200020085	200020085@iitb.ac.in
Akshata Koshti	200070034	200070034@iitb.ac.in
Vaishnavi Agnihotri	200020155	200020155@iitb.ac.in
Siddhi Gaikwad	200110037	200110037@iitb.ac.in
Adit Agrawal	200040009	200040009@iitb.ac.in

Table of Contents

Inspiration for Idea

The primary motivation was to help paralyzed people by giving them a sense of entering into a new body to control completely. They cannot perform the actions independently because the link between their brain and body parts is disrupted, but we could certainly think of something that they can control only using their thoughts!!!
We aimed to make a bot that could operate using brain signals. So the human has to think of the action which needs to be performed, and the robot would execute the same.

Problem Statement

Create a robot that can perform the hand movements as imagined by the person from whom EEG signals are recorded.
This goal includes two main problem statements as follows:

1. Extracting the exact command/ activity from EEG data.
2. Creating a bot and synchronizing the output from EEG data for being able to move a bot.

Existing solutions in the Market

Current solutions include just one of the artificial body parts to be accessible via EEG signals. Like a

- Robotic arm
- Robotic leg
- Automatic Window blind
- Smart wheelChair

Proposed Solution

Our mission is to create a complete humanoid robot and not just one part of it, which can be entirely controlled by thoughts. To be more Technical, we propose to take EEG signals from the scalp using non invasive technology, process it using a deep neural network and send the signals to a robot, thereby controlling it. This will definitely help the paralyzed. We also propose that the building material of the robot can be changed and it can be sent to any place which is not tolerable for Human beings.

Brief Description

The project aims to use brain signal datasets (EEG Datasets) that control basic hand movements like picking and dropping an object to operate a hand robot. The workflow included searching for appropriate datasets and finding ways for how Bot could use them. The current project makes a simulation of the bot rather than building it physically. The human brain contains different parts to control various parts of the body, this control is done using neural signals. So the plan is to track the electric signals produced by those neural signals from the cortex by EEG technology and interpret them so that the bot could use them to perform desired actions.

Progress

Workflow

Firstly, all team members studied EEG in detail. Then we split our work into 2 modules.

Module 1 work include:

Before review meet 1	1. **Research more on EEG and collect datasets** : Researched a lot on EEG. Collected datasets and selected dataset on hand movements from Kaggle.This data contains EEG recordings of subjects performing grasp-and-lift (GAL) trials. It contains a raw dataset and corresponding results. **2. Learnt Python, Matlab, EEGLAB** Learned MATLAB from onramp with a special emphasis on EEGLAB and spent a good time with it to understand how EEG signals work and how it is filtered. Learned basics of python.
Before review meet 2	1. Learned **basics of machine learning** and deep learning 2. **Trained a model** found on Kaggle with fine tuning according to the requirements of Bot. 3. Made a **GitHub repository**.
After review meet 2	**1. Tried to gain more accuracy** tried alternative pathways, like doing a 7 class classification and replacing sigmoid layers with Softmax. But we didn't get any better results **2. Tried for live data** Analysed the problems which can be encountered while trying the model on live data. Tried to resolve those problems. Planned to perform the experiment. **3. Combining two modules** Made a code such that it can be given to ROS as an input.

Module 2 work includes :

Before review meet 1	**1. Learn more about Ubuntu and Linux interface** We were not very familiar with the Ubuntu environment as we used Windows, so we spent time configuring the Linux interface on the virtual box. **2. Solidworks** Tutorials, installation, and basic stuff through cad cam tutorials cause we first planned to develop a bot of our own, but we were time-constrained.

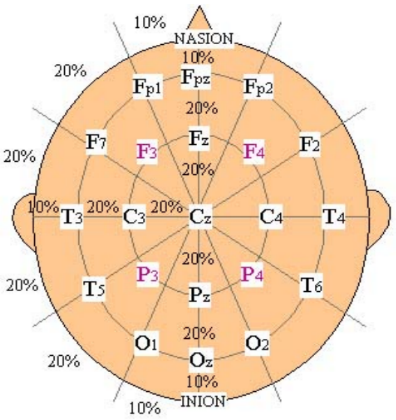
	3. **Turtlebot installation.** We used turtlebot as it is one of the most basic bots, which allowed us to take a deep dive into robotics.
Before review meet 2	1. **TurtleBot SLAM and Navigation**: Used Gazebo and RVIZ for the first time 2. **ROS beginner and intermediate level tutorials**: This helped us get a great idea about robot operation and simulation, creating and building packages, etc. 3. **Learn Python**: To manipulate the code given by module_1, we needed some knowledge about python. 4. **Open Manipulator**: We installed 3 manipulators and practiced SLAM, navigation, arm, and gripper movements.
After review meet 2	1. **Execution of simple codes on manipulator**: We searched for codes based on gripper and arm motions on GitHub and executed them. 2. The final task was to **collaborate with Module 1** to edit the code to be used by the manipulator to produce desired actions. 3. **Debugging**: It was the last and the most time acquiring task! It involved installing PyTorch, matplotlib, NumPy, etc, so that the ROS environment may become favorable to use EEG code.

CHALLENGES FACED

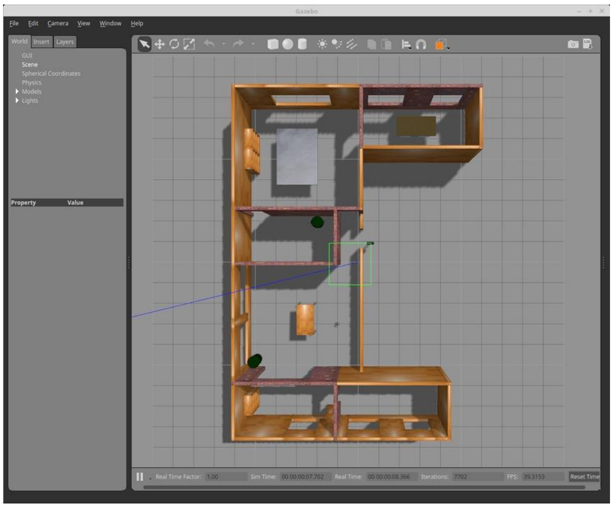
- Collecting relevant datasets of EEG
- Selecting the BCI (EEG/MEG)
- Exact mapping to the electrode to site
- Training the neural network model on large size of data (frequent RAM crashes)
- Tried developing a model which can work on Live data but due to time constraint we were unable to perform the experiment
- Tried for moving open manipulator but failed coz of time constraints.
- Used Ubuntu 20.4 instead Ubuntu 16.04. This gave a huge setback to module 2 as 70% of the work was done on ubuntu 20.04 and we realized quite late that we needed ROS Kinetic, so that killed a lot of time!
- Frequency for Turtle bot and PyTorch were different.

Results

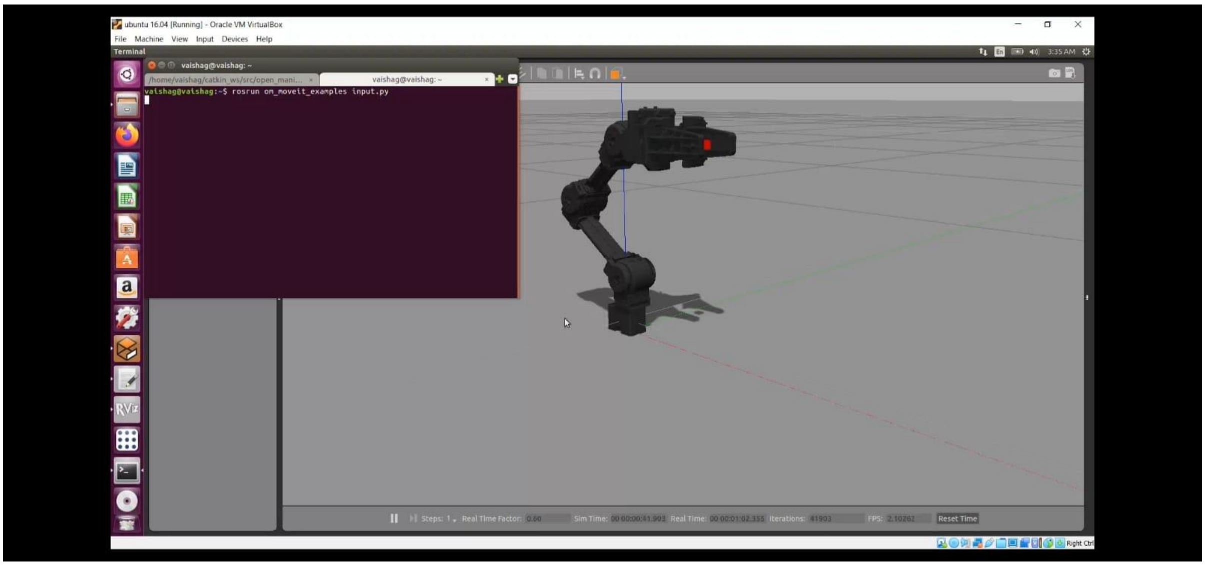
Final PPT	Final Presentation by Team Inception
Video	[ROS Demo.mp4]
Github Repo	[Github]
Other Docs	List of datasets Review meet 1 report: Team Inception Review meet report 2: Team Inception



10-20 System

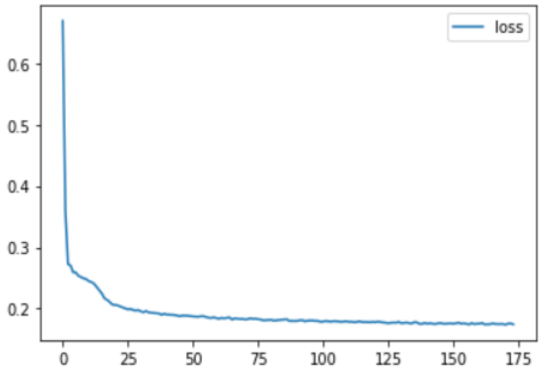


Gazebo Waffle_Pi Environment

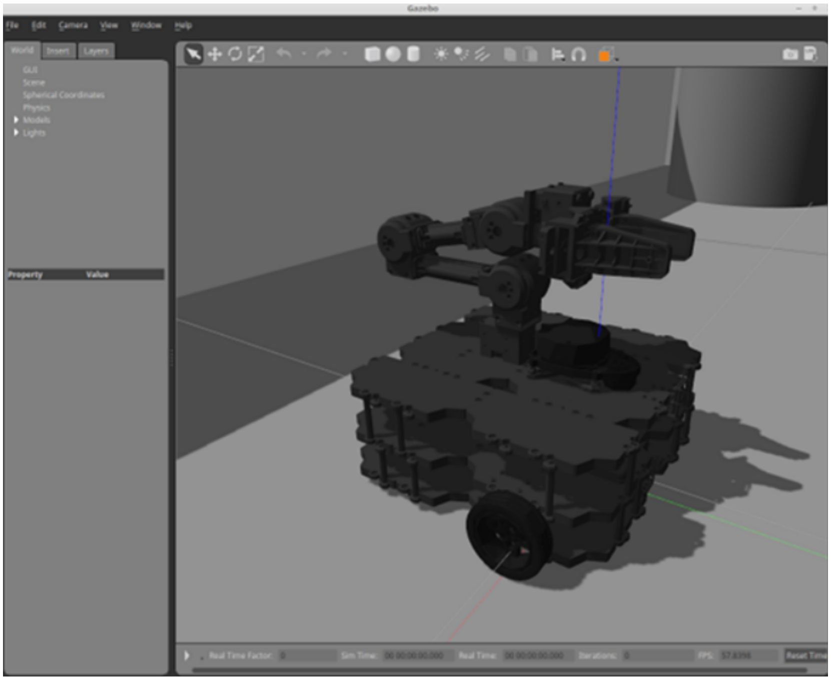


```
vaishag@vaishag:~$ rosrun om_moveit_examples input.py
```

Final Output



Loss Function



Open Manipulator with TB3

Learning Value

- Learnt **Python** and **MATLAB**
- Went through **Basics of machine learning**
- Got trained in **Deep learning**
- Got to know about **Configuring Ubuntu environment**
- Worked more on **ROS and Turtlebot**
- Gained insights of **Open Manipulator Simulation**
- Combined the deep learning model and working of hand robot
- Gained experience of how to work in a team
- Enhanced **Time management & Planning skills.**
- **Researching and contacting** with seniors/experts

Software/ Hardware used

Softwares:

- MATLAB
- ROS
- GAZEBO
- PYTHON
- SOLIDWORKS
- TURTLEBOT3 WAFFLE_PI
- OPEN MANIPULATOR-X
- OPEN MANIPULATOR WITH TB3
- VIRTUAL MACHINE
- UBUNTU 16.04
- GOOGLE COLAB
- JUPYTER NOTEBOOK
- ROS KINETIC KAME
- FRAMEWORK: PyTorch
- EEGLAB

Suggestions for others

1. The most significant setback faced by our project was working on the wrong ubuntu version. We first installed Ubuntu 20.04 which is the latest version, but it is not compatible with open manipulator. So that created a lot of chaos.
2. Get hold of an EEG headset before starting with the project. This will help a lot to create live data.
3. Creation of 2 Modules really helped us a lot to complete a wide project within time. It can be said as one of the best decisions of this team project. So we further suggest other teams to practice work division.

Contribution by each Team Member

Om Mihani (Team Leader)

- **Planning of the project:**
  - Made the original idea of the project and distributed the works amongst the team members. Also, did overall supervision and coordination.
- **Contacting:**
  - Did all the “consulting the seniors and professors” part for EEG.
- **Research:**
  - Along with others, did a good amount of study on EEG and searched for proper datasets.
- **MATLAB:**
  - Learned EEGLAB of MATLAB to understand EEG signals and their filtering.
- **Machine Learning and Deep Learning:**
  - Took the above said tutorials from coursera to train the model.
- **Live data planning:**
  - Tried and contacted a lot of seniors for an EEG live data experiment.
- **Training of neural network:**
  - Trained the final model to be used in the code.
- **Accuracy check:**
  - Tried other neural networks to get a better accuracy.
- **Module integration:**
  - Made suitable codes such that the python module could be coupled with the ROS module.

Akshata Koshti

- **Learning Python**
  - Learned python from book and past year learner space resources for better understanding of ML.
- **Learning Machine learning and deep learning**
  - Learned the basics of machine learning and deep learning.
- **Training of the neural network**
  - Trained a pre trained model of neural network
  - Modified the code according to the requirements of the bot
- **Trying different type of neural network**
  - Tried building a neural network by replacing sigmoid function by softmax for obtaining better results.
  - Trained a new neural network and checked accuracy.
- **Content creation**
  - Contributed in the content of GitHub readme and abstract of the project.
  - Preparation of presentation and documentation.

Vaishnavi Agnihotri :

- **Completed ROS Beginner and Intermediate Level Tutorials**

ROS tutorials From ROS WIKI. It included learning creation of packages, nodes, publisher, subscriber, operating turtlesim and use of rqt graphs etc.

- **TurtleBot Tutorials**

  It included installation of TURTLEBOT3 WAFFLE_PI and practicing stuff like SLAM, navigation and translation of Bot.
- **Open Manipulator**

  OM tutorials through E- Robotis. Installed 3 manipulators in total and executed moveit codes from github in order to check gripper and arm movements of hand robot.

- **Solidworks**
  Installed Solidworks, practiced CAD CAM tutorials which included creation of screws, bearings etc. Also practiced through ERC workshops and ME119 Lecs.
- **Python**
  Basic Python Learning through Coursera(Python for Everybody) and PYCK
- **OM Code :**
  Modified the node code to move the manipulator via functions.
- **Final Video :**
  Compiled all the codes and recorded the final video and did the editing along with its soothing music.
- **Content Creation**
  Contributed in the content of Github Readme and abstract of the project.

Adit Agrawal **:**

- **ROS Beginner and Intermediate Level Tutorials**
  ROS tutorials From ROS WIKI. It included learning creation of packages, nodes, publisher, subscriber, operating turtlesim and use of rqt graphs etc. Also gathered more knowledge through ERC Workshops.

- **Python**
  from Coursera : Python 3 Programming

- **TurtleBot Tutorials**
  It included installation of TURTLEBOT3 WAFFLE_PI and practicing stuff like SLAM, navigation and translation of Bot.
- **Open Manipulator**
  OM tutorials through E- Robotis. Installed 3 manipulators in total and executed moveit codes from github in order to check gripper and arm movements of hand robot.

- **Open Manipulator Code :**
  - Modified the node code to move the manipulator via functions. .

- **Interfacing :**
  - Manipulated the output of the EEG code so that it could be used by open manipulator to perform the tasks.
- **A Step Further :**
  - Tried for translation of manipulator with Turtlebot 3
  - To make the gazebo environment more appealing

Siddhi Gaikwad**:**

- **Python:**
  - Got trained in python to aid the ML module
- **Github :**
  - Took the entire responsibility of maintaining the github Repo.
- **Documentation :**
  - Did a major role in documentation and content arrangement.

References and Citations

- AUTOMATE THE BORING STUFF WITH PYTHON (Author- A L Sweigart)
- wncc/learners-space
- iMotions_EEG_Guide__2019 (2).pdf
- Coursera ML
- Coursera DL
- EEGLAB News
- Welcome to the EEGLAB Wiki - EEGLAB Wiki
- FieldTrip toolbox: Welcome to the FieldTrip website
- MNE — MNE 0.23.0 documentation
- ClassificationAndAnalysisOfEEGSignalsIITD.pdf
- EEG differentiates left and right imagined Lower Limb movement.pdf
- eeg-controlled-smart-wheelchair-for-disabled-people (1).pdf
- Fundamental_of_EEG_Measurement (1).pdf
- Hybrid EEG.pdf
- LR hand movement.pdf
- Mapping of EEG signals by Dr. Tapan Gandhi.pdf
- PyEEG.pdf
- Schirrmeister's paper Deep learning with convolutional neural networks for EEG dec…
- The leg mvement EEG.pdf
- EEG Motor Movement/Imagery Dataset
- Data sets
- Kaggle
- EEG data from hands movement
- meagmohit
- BCI Competition III - Data Set
- ROS Robot Programming : A Handbook Written by TurtleBot3 Developers
- ROS Robotics by example : Second Edition by Carol Fairchild
- Interfacing of EEG with Manipulators
- Python 3 Programming Specialization Course
- Turtlebot3
- IMU Basics .
- Solidworks Playlist
- ROS Tutorials
- Open Manipulator-X Tutorials
- om_moveit_examples
- open_manipulator_with_tb3
- CAD CAM Tutorials

Disclaimer

FAIR USE DISCLAIMER:
This project may contain copyrighted material, the use of which has not been specifically approved by the copyright owner. We believe that our use of such material falls strictly within the guidelines of fair use.

COPYRIGHT DISCLAIMER:
The material used in this project and the manner of use lies well within the terms of the respective copyright policies of the resources used.

Licenses

PyTorch
The core of ROS is licensed under the standard three- clause BSD license.
Licensed through LDAP:

- Solidworks
- MATLAB

Open Source

- Google Colab
- Gazebo is open-source licensed under Apache 2.0