*Fuzzy Systems and Data Mining IX*
*A.J. Tallón-Ballesteros and R. Beltrán-Barba (Eds.)*

*© 2023 The authors and IOS Press.*
*This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).*
*doi:10.3233/FAIA231095*

# Portable Brainwave Controlled Robotic Dog for Therapy

Chi-Yat Lau [a], Man-Ching Yuen [b,1], Chi-Wai Yung [b], JiaJie Liang [a], Wing-Chi Mak [a], Ho-Yeung Chan [a], Fong-Yu Ha [a], Zheng-Tan Ng [a] and Hong-San Wong [c]
[a] *Department of Information Technology, Vocational Training Council, China*
[b] *Department of Applied Data Science, Hong Kong Shue Yan University, China*
[c] *R2C2 Limited, China*

**Abstract.** With the revolution of technology, robots are an inevitable trend as robots can do what humans do more efficiently and at a lower cost. However, humans have retained labor force dominance because robots have suffered from immature software and hardware. To solve the problem of immatureness, we exploit a platform that can integrate the systems of multi-robots and strives to let all the users control their own robots through the platform. Our goal is to develop a ROS (Robot Operating System) package that can incorporate a small robot dog which allows the robot dog to respond to our industrial partner, R2C2, webapp commands and in addition to using brain waves to control the walking path. Our platform provides an intuitive and accessible interface for users to control the robot's movements and behavior, increasing the robot' s versatility and usage for applications such as therapy, entertainment, education, and research. Overall, our project represents an exciting step forward in the field of robotics, and we are excited to continue exploring new ways to improve the interaction between humans and machines.

**Keywords.** EEG, brain wave control, robot

## 1. Introduction

### *1.1. Background*

In an era where technology is rapidly advancing, robots have emerged as a powerful tool that can help humans in numerous ways. Robots are used in different applications such as teaching and learning [1]. With this in mind, our research project aims to provide a comprehensive platform that unites the control of robots, making it easier for users to control and interact with these machines. Our project is being developed in collaboration with our industrial partner, R2C2, who has provided us with a website platform that helps us integrate our work with their platform. R2C2 is a Hong Kong-based robotic startup that focuses on construction and agriculture automation.

To explore new possibilities for robot control, we use brainwave control, which will enable us to control the robot using our thoughts. To achieve this, we use an EEG (Electroencephalography) [2] sensor, which will pick up signals from our brain and translate them into commands that the robot dog can understand. We believe that this

[1] Corresponding Author, Man-Ching YUEN, Department of Applied Data Science. Hong Kong Shue Yan University, Hong Kong, China; E-mail: mcyuen@hksyu.edu.

new level of control will not only make it easier for users to operate the robot dog but also opens new possibilities for how we interact with machines. By tapping into our brain's natural abilities, we can create a more intuitive and seamless way of controlling robots, making them more accessible to a wider range of users.

### *1.2. Our Contribution*

The project aims to develop a comprehensive and functional ROS (Robot Operating System) package for a small robot dog which can be controlled by human brain waves. Our contribution includes:

- This package enables the robot dog to respond to commands from the R2C2 web application and control its walking path using brain waves. The package is designed with a clear and detailed software architecture that includes communication protocols and control mechanisms to ensure the robot operates effectively to provide users with a versatile and easy-to-use platform to control their robot.
- We conduct extensive testing and validation to ensure that the ROS package functions correctly, meets all the required specifications, and is robust and reliable.
- We provide clear documentation and user guides to enable users to easily install, configure, and use the ROS package, reducing the complexity of the startup process for the robot.

The organization of this paper is as follows. Section 2 describes the background and related work. Section 3 presents the methodology used in our proposed system. Section 4 shows the solution design. Section 5 presents the system testing and performance analysis. Section 6 draws out the conclusion.

## 2. Background and Related Works

### *2.1. Robot Operating System (ROS)*

The original aim of creating the Robot Operating System (ROS) was to develop a stable and versatile robotics software platform. ROS is an open-source meta-operating system that provides core operating system services such as message-passing between processes and package management. It already includes a wide range of drivers, state-of-the-art algorithms, and robust developer tools. ROS simplifies the process of building stable and complex robot behaviors on various robotic platforms by providing a set of tools, libraries, and conventions. The ROS Filesystem can streamline the project's build process while allowing for flexibility in managing dependencies [3]. Our robotics project can benefit from ROS's comprehensive features and capabilities.

Figure 1 shows the ROS conversation. In ROS, topics serve as named communication channels that enable nodes to exchange messages. They employ anonymous publish/subscribe semantics, which decouples the generation of information from its consumption. Topics are designed for unidirectional, streaming communication, and are not suitable for nodes that require bidirectional communication, such as remote procedure calls - in such cases, services should be used instead. Our project utilizes topics to facilitate communication between the robot and the platform, allowing the robot to make calls and receive responses that inform its actions.

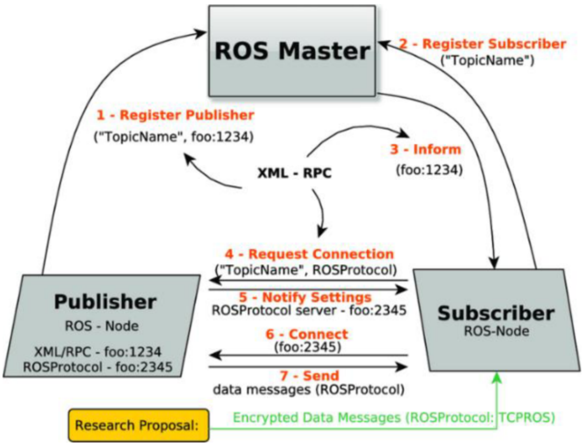


**Figure 1.** ROS Conversation

## *2.2. Electroencephalography (EEG)*

Electroencephalography (EEG) [2] is a technique used to map the spontaneous electrical activity of the brain by measuring the electrical signals generated by pyramidal neurons in the neocortex and allocortex. Figure 2 shows the EEG concept [2]. This method typically involves placing noninvasive EEG electrodes along the scalp, often using the International 10-20 system or similar variations.

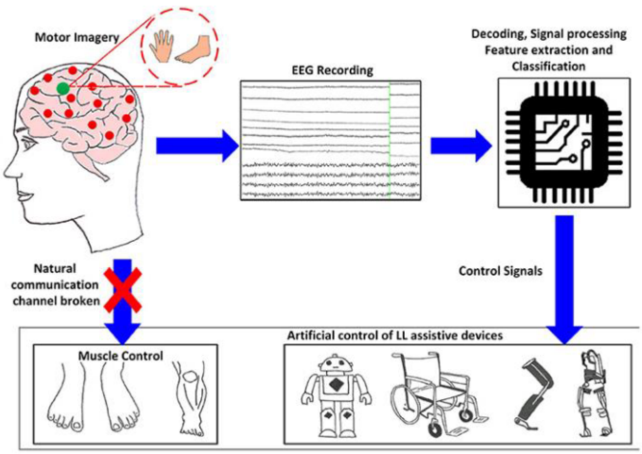


**Figure 2.** the EEG concept [2]

EEG recordings are commonly interpreted either visually by examining the traces or through quantitative EEG analysis. This technique is widely used in clinical settings for diagnosing and monitoring neurological conditions. EEG is a noninvasive procedure that provides valuable insight into the electrical activity of the brain, making it a valuable tool for understanding brain function and for diagnosing and treating neurological disorders.

EEG data analysis is a unique approach to studying cognitive processes that has diverse applications. Medical practitioners can use EEG data analysis to make accurate diagnoses, while researchers can explore how brain processes shape human behavior.

### *2.3. Related Work*

There are some existing works on studying EEG and robot interaction, such as [4]. However, these systems did not have an easy-to-use interface, so it can only be used for demonstration in the laboratory. In this project, we develop an easy-to-use interface and our system is a portable solution, so it is can be used anywhere.

## 3. Methodologies used in our Proposed System

In the project, we develop a comprehensive and functional ROS (Robot Operating System) package for a small robot dog which can be controlled by human brain waves. In our Robot Operating System (ROS) package, we add Web ROS Proxy and Minipupper Control to receive and process information from the Web App.

### *3.1. Web ROS Proxy*

We use Web ROS Proxy in Minipupper to receive and parse the data from the Web App via socketio. After receiving the data from socketio, trigger_event will split the data. This is to make it easier to distinguish between different message types, because the msg type must be the same between two topics to communicate with each other. This allows Minipupper control to send the data to the corresponding topics smoothly.

### *3.2. Minipupper Control*

The Minipupper control is mainly responsible for how these messages should be handled in the catkin workshop after receiving the Web ROS Proxy message. As the request is received, the topic handler will call out the functions that are related and send data to make movement.

### *3.3. Minipupper Bringup*

Based on the open source project CHAMP, the Bringup package is mainly responsible for the operation of the basic functions of Minipupper. It allows Python programs to connect to Minipupper's hardware for deployment. This is the main package that controls the four motors and twelve joints of Minipupper, and it handles the operation of the four motors by subscribing to the relevant topic message and calling the callback.

### *3.4. Systemd Module*

We use the *Systemd* module to create different services for the robot dog. It enables the robot dog to automatically listen to multiple topics at the same time when it is booted, and to handle multiple programs at the same time. Eventually, Minipupper will be able to initialize the base package and control it directly from the Web App when booting Ubuntu.

### 3.5. Camera

To remotely control a robot dog, you typically need to have access to information about the robot's current environment, such as its location, orientation, and any obstacles that may be present. This information can be obtained through various sensors and systems that are built into the robot. In order to remote control the robot dog, we also need to know the current environment of the robot dog. In this case, we are equipped with cameras. Cameras are a valuable tool for remote control of robotic dogs because they provide a visual representation of the robot's environment. Moreover, cameras can provide real-time video feedback.

### 3.6. EEG: Convert data into digital data for data analysis by using machine learning methods

To convert EEG data into digital data for data analysis, we use several libraries, including ssqueezepy, timm, pytorch-lightning, mne and pylsl.

In the following, we describe the steps of EEG data signal processing.

**Step 1: Read CSV Data.**

- Figure 3 shows the international 10-20 system [5]. Extract the natural, forward and turn left EEG data from the emotiv. Read three CSV files into three separate data frames named ‘df0’, ‘df1’, and ‘df2’. After a series of data concatenation operations, a new data frame is generated.

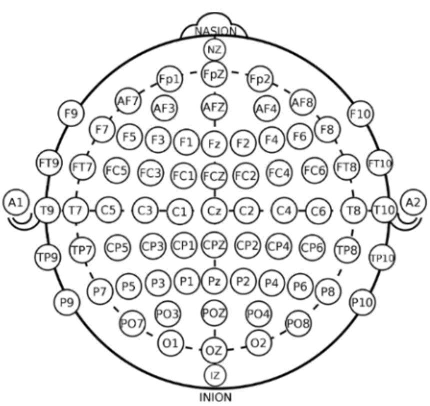


**Figure 3.** International 10-20 system [5]

**Step 2: Convert each data from CSV file to mne format.**

- We segregate the subject based on the ‘group_id’ and ‘subject_id’ so we create a group by object which creates a group on each ‘group_id’ and ‘subject_id’. We pass the ‘grp1’ variable to the ‘convertDF2MNE’ function, convert the data frame to mne and check the shape. There are 97 epochs, 5 channels and each epoch has a length of 384.

**Step 3: Convert signal to image.**

- We pass 5 channels and 384 sequence length data to the continuous wavelet transform function. We use a function called 'imshow' from 'ssqueezepy' to plot the scale gram image. We can see our scalogram image in Figure 4.

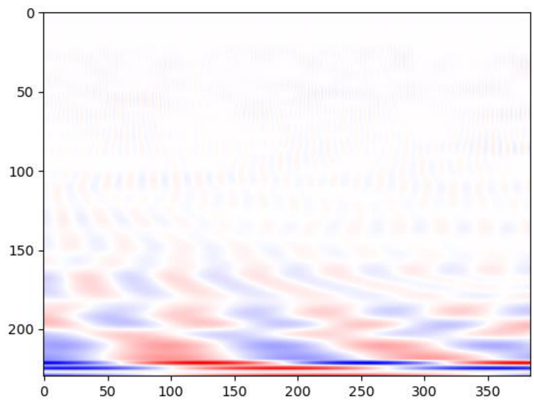


**Figure 4.** Convert signal to image

**Step 4: Iterate all groups one by one.**

- After computing the wavelet transform, the next step is to save the resulting images for each trial in a separate directory. We use a loop that iterates through each group of MNE data and each trial within that group. For each trial, the wavelet transform is computed, converted to an absolute value, and saved as a numpy array in a directory specific to the corresponding subject and trial.

**Step 5: Train a pre-trained ResNet model.**

- We define a PyTorch Dataset class called 'DataReader' for efficiently loading and processing the data during training and validation. Since we can't pass all of the images at once. We preprocessed the images and randomly selected images for the flip operation.
- First, we split the dataset into training and validation sets using the 'random_split' function from the torch.utils.data module. Then, we initialize the 'OurModel' class to create the neural network model. Next, we initialize the 'Trainer' class with various parameters that determine how the model will be trained, such as the maximum number of epochs, the learning rate, and the precision of the computations. The 'Trainer' class also specifies that the training will be done on a GPU if one is available. Finally, the fit method is called on the 'Trainer' object to train the model.

## 4. Solution Design

In this section, we present the design of our solution – a ROS (Robot Operating System) package that can incorporate a small robot dog which allows the robot dog to respond to our industrial partner, R2C2, webapp commands and in addition to using brain waves to control the walking path.

### 4.1. System Architecture

In Figure 5, our system is mainly divided into three layers by using socket.io and the ROS publisher passing through the request and response. By using the webapp or the EEG EMOTIV data and signal will be sent to the Web ROS Proxy inside the Minipupper ROS through the socket.io and the data and signal will be processed by the Web ROS Proxy and publish to the Minipupper file system catkin and execute by the Minipupper control package and the Minipupper control package call out and send data to the champ_controller and achieve the movement. Table 1 shows the hardware requirements of our system. Table 2 shows the software requirements of our system.

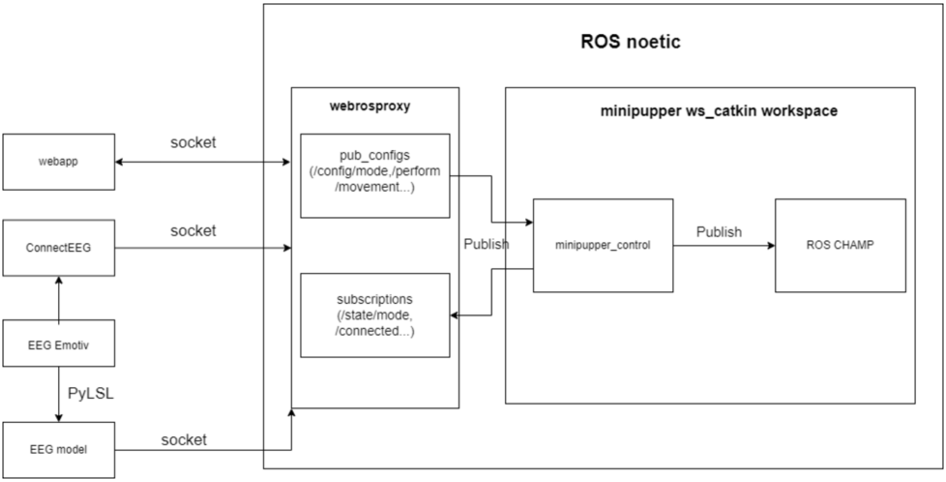


**Figure 5.** Architecture of our system (3-layer design)

**Table 1.** Hardware Requirements

| | Required Hardware | Requirement |
|---|---|---|
| **1** | Robot Dog | A ROS based system robot |
| **2** | Raspberry Pi | SD card with Ubuntu 20.04 and ROS installed |
| **3** | EEG Sensor | A sensor to detect our brainwaves |
| **4** | Computer | A computer that can run MacOS/Windows. |

**Table 2.** Software Requirements

| Component | Required Software |
|---|---|
| **Computer** | Ubuntu 20.04 |
| | Ros noetic |
| | Python 3.6 |
| **Raspberry Pi** | Ubuntu 20.04 |
| | Ros noetic |
| | R2C2 Web App |
| | Web Ros Proxy |
| | Cortex API |

### *4.2. Testing Module*

Our testing is based on the below module:

**Minipupper**

- Minipupper is a dog-shaped robot sidekick for learning and experimenting with robotics, exploring advanced functions of a dynamic robot. Figure 6 shows Mangdang Minipupper.

**EEG sensor**

- To implement the EEG control a EEG controller installed with the API is essential. Additionally, the laptop equipped with Bluetooth is used to control EEG API to record EEG samples.
- The chosen EEG sensor is shown in Figure 7.
    - o Product: Emotiv Insight
    - o Channel: 5 channels
    - o Shape: wireless headset

**Figure 6.** Minipupper

**Figure 7.** Emotiv Insight

### *4.3. Target Users*

The EEG robot dog is a relatively new technology, and its target user can vary depending on the specific application and intended use case. However, in general, the EEG robot dog is often designed and marketed as a therapeutic tool for individuals with autism, ADHD, or other neurodevelopmental disorders. The robot dog is equipped with an EEG sensor that can detect brain waves. When a patient interacts with the robot dog, the doctor will go to observe the patient's brain waves, and then process them through machine learning algorithms to interpret the user's emotional and cognitive state. This feedback can be used to provide personalized therapeutic interventions, such as calming exercises or social skills training, to help improve the user's overall well-being and quality of life. Additionally, the robot dog can also be used as a research tool in neuroscience and psychology to better understand the brain's response to different stimuli.

### *4.4. Limitations*

There are a number of limitations in this project which are shown as follows:

- *Battery Life*: We find that if we add all the plugins to the robot the default battery of the Minipupper is not enough to keep moving the robot for 30 minutes, the motor will not have enough power to push up the Minipupper. It is hard to test Minipupper movements. We are going to try to link to another battery to increase the battery life. We also need to decide on all the plugins we need for our project.

- *Sample data*: We cannot find any sample data from the internet, so we need to enter the 70% training and 30% testing split to train our machine learning.
- *Low Sensitivity*: The EEG boasts only produce 5 channels of EEG sensor which is not enough for observing the changes of the brain waves, so the sensitivity of the boasts is lower than ideal. Due to the low sensitivity, 3 directions can only be made and detected successfully.

## 5. System Testing and Performance Evaluation

We successfully establish a connection between the Minipupper and the R2C2 webapp. Next, we collect data on the movements and behaviors of the Minipupper and use the collected data for model training. Finally, we integrate an EEG sensor with the Mini Pupper and R2C2 webapp, develop software to interpret the signals from the EEG sensor and use the interpreted signals to control the movements of the Mini Pupper through the R2C2 webapp.

Figure 8 and Figure 9 show the system testing result. The test set accuracy was **86%**, which is a decent performance. The classification report shows precision, recall, f1-score, and support for each class. The macro-average F1-score is **0.79**, indicating that the model has a fair ability to classify three classes.

```
              precision    recall  f1-score   support

           0       0.78      0.44      0.56        16
           1       1.00      0.93      0.96        42
           2       0.76      1.00      0.86        31

    accuracy                           0.87        89
   macro avg       0.84      0.79      0.79        89
weighted avg       0.88      0.87      0.86        89

sklearn 0.8651685393258427
```

**Figure 8.** Test accuracy

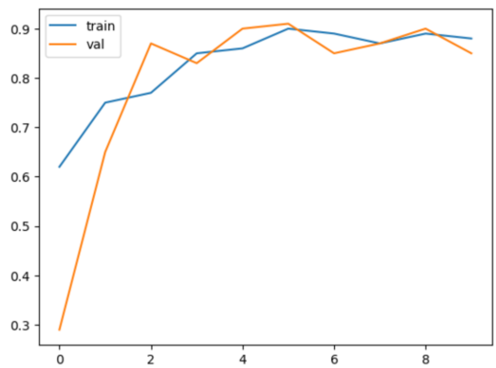


**Figure 9**. Test model image

## 6. Conclusion

Our project provides a significant insight to the future of robotic design in commercial products and research topics. The outcome of this project is the successful development and implementation of a ROS package that enables a small robot dog to respond to commands from the R2C2 web application and control its walking path using brain waves. This will provide users with a versatile and easy-to-use platform to control their robot, reducing the complexity of the startup process and increasing the robot's usage and versatility. Specifically, the expected outcomes are: (1) The ROS package undergoes a robust and reliable implementation process, including extensive testing and validation to ensure it meets all necessary specifications and functions effectively under various conditions and scenarios. The implementation process ensures that the package is user-friendly and easily accessible. (2) Clear and comprehensive documentation and user guides will be provided to assist users in the installation, configuration, and use of the ROS package. This documentation will reduce the complexity of the startup process for the robot and enable users to easily operate the robot according to their needs.

Overall, successful completion of this project results in a versatile and user-friendly platform for controlling a small robot dog. This platform provides an intuitive and accessible interface for users to control the robot's movements and behavior, increasing the robot' s versatility and usage for applications such as entertainment, education, and research. We use Minipupper for testing the connection, in the future we will be transporting our coding to other robots to achieve robot mission collaboration. Moreover, we will conduct more experiments for comparison with state-of-the-art fusion techniques.

## Acknowledgements

This research was in part supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. UGC/FDS15/E02/20 and UGC/FDS15/E01/21).

## References

[1] Yuen M. C., Chan K. K., Li K. S. (2021). Mobile App Controlled Modular Combat Robot for STEM Education. Proceedings of 2021 13th International Conference on Communication Systems & Networks (COMSNETS), Workshop on Machine Intelligence in Networked Data and Systems (MINDS), IEEE CS Press, Webinar, January 2021.

[2] Tariq Madiha, Trivailo Pavel M., Simic Milan (2018). EEG-Based BCI Control Schemes for Lower-Limb Assistive-Robots. Frontiers in Human Neuroscience 12.

[3] Anggraeni, Pipit. (2019). Decentralized leader-follower consensus for multiple cooperative robots under temporal constraints.

[4] Mao, X., Li, M., Li, W., Niu, L., Xian, B., Zeng, M., & Chen, G. (2017). Progress in EEG-based brain robot interaction systems. Computational intelligence and neuroscience, 2017.

[5] The International 10-20 system. https://zh-yue.wikipedia.org/wiki/10-20_%E7%B3%BB%E7%B5%B1