Error-related potentials-based human-robot intelligent system

Xiaofei Wang

School of Computer Science

Faculty of Engg. & IT

University of Technology Sydney

NSW - 2007, Australia

Error-related potentials-based human-robot intelligent system.

by Xiaofei Wang

A thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy
in
Software Engineering

Principle Supervisors: Professor Chin-Teng Lin

Co-Supervisors: Dr. Tim Chen, and Dr. YK Wang

School of Computer Science
Faculty of Engineering and Information Technology
University of Technology Sydney

February 2022

Certificate of original authorship

I, *Xiaofei Wang* declare that this thesis, submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the *Computer Science*, *Faculty of Engineering and Information Technology* at the University of Technology Sydney, Australia, is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution. This research is supported by the Australian Government Research Training Program.

Production Note:
Signature removed prior to publication.

SIGNATURE: ______________________

[Xiaofei Wang]

DATE: 4th July, 2022

PLACE: Sydney, Australia

Abstract

Brain-Computer Interface (BCI) is an emerging technology that provides natural and direct communication between humans and machines. Recent BCI works aimed to create accurate and reliable BCI systems in the field of Human-Robot Interaction (HRI). Of these, the BCI paradigm based on error-related potentials (ErrPs), a cognitive phenomenon derived from EEG signals, is particularly promising. ErrPs are involuntarily evoked when a person perceives unexpected errors in the environment. Unlike other BCI paradigms that require users to actively imagine the mental commands or engage with additional visual stimuli, ErrPs depends on the user's experience on assessing the correctness of the robot behaviours. The ErrP-based BCI does not require additional training and does not interrupt the user's original workflow. This thesis presents two novel ErrP-based BCI systems:

First, a novel robotic design for ErrP-based BCI that allows humans to evaluate the robot's intentions continuously. Current ErrP-based BCI cannot handle interaction sequences that involve continuous robot movements. For example, it is difficult to extract a time-locked event when the user detects an unexpected error while the robot arm is already in motion. The high classification accuracy (77.57%) from the first system confirmed that the proposed ErrP-based BCI paradigm allows continuous evaluation of robot intentions in real-time and thus enable earlier intervention before the robot commits an error.

Second, ErrP-based shared autonomy via deep recurrent reinforcement learning.

Current BCI systems use ErrP as either an implicit control signal to the agent or a reward signal in reinforcement learning (RL). Our novel framework proposed using ErrP as an input feature in the trained RL model, which enables human intervention with a trained autonomous agent. In a simulation with 70% ErrP accuracy, agents completed the task 14.1 % faster. In the real-world experiment, agents completed the navigation task 14.9% faster. The evaluation results confirmed that the shared autonomy via deep recurrent reinforcement learning is an effective way to deal with uncertain human feedback in a complex HRI task.

These two novel BCI systems advance the current ErrP-based BCI capabilities and enable a wide range of new interaction possibilities between human and robot. This thesis represents an important step toward a BCI-based shared autonomy between humans and robots.

Acknowledgments

I want to express my sincere gratitude to my principle supervisor, Professor CT Lin, for the guidance, motivation and weekly feedback during these years. He has always motivated and guided my work in the right directions. Without those, I don't think I would have been able to finish this project. It is my great honor to be supervised by Professor CT Lin. I would also like to thank my co-supervisor, Tim Chen. Thank you for the hard work and endless support. Whenever there is a problem in research or outside the school environment, he always tries his best to help. Thank you for all the help, guidance, and mentoring received during these years. I would also like to thank my co-supervisor, Dr. YK Wang. Thank you for the weekly discussion. Thank you for the help on papers and thesis.

I want to thank all the CIBCI lab members for all the help and knowledge they provided during my years. Special thanks to Carlos Tirado, Fred Chang, Tien-Thong Do, Avinash Kumar Singh, Howe Zhu, Jia Liu, Jie Yang, Sai Kalyan Ranga, Yanqiu Tian, and Liang Ou.

I want to thank all my friends in Techlab. Thank all the friends since the first day I came to Techlab. And thank all the friends outside the school environment. The green mountain won't change, the flowing water is endless. See you around!

I want to thank the Australian Research Council (ARC) for its financial support.This work was supported in part by the ARC under discovery grant DP180100656 and DP210101093. I also want to thank the UTS International Research Scholarship for

covering my tuition fees.

Last but not least, I want to thank my family for their constant love and support. Without their support and motivation, I would not be where I am now. No words can express how thankful and grateful as being one of the members in my family. I love you all!

DEDICATION

To my father, my mother, and my brothers for their endless love, support and encourage . . .

List of Publications

Journal :

1. **X.-F. Wang**, H.-T. Chen, Y.-K. Wang, C.-T. Lin, "Implicit Robot Control using Error-related Potential-based Brain-Computer Interface," IEEE Transactions on Cognitive and Developmental Systems. (accepted)

2. **X.-F. Wang**, H.-T. Chen, Y.-K. Wang, C.-T. Lin, "Error-Related Potential-Based Shared Autonomy via Deep Recurrent Reinforcement Learning". (draft)

3. **X.-F. Wang**, H.-T. Chen, Y.-K. Wang, C.-T. Lin, "Error-Related Potential used to correct robot movement during continuous searching task in a maze". (draft)

Table of Contents

LIST OF FIGURES

FIGURE Page

List of Tables

Chapter 1

Introduction

1.1 Motivation

BCI enables humans to interact with external devices by using control signals generated from electroencephalographic activities without muscles. BCI opens a new channel for humans to interact with their surroundings. This can make the brain signal communicate directly with external systems. BCI can be invasive and noninvasive according to whether implanting electrodes into the cortex. The noninvasive BCI has been more frequently used such as EEG MEG. Recently, noninvasive electroencephalogram (EEG), which measures brain signals with electrodes put on the scalp, was commonly used to control robots, such as state visual evoked potentials (SSVEP), Motor Image (MI), P300 potentials. Millan [48] used MI to move a cursor towards a target successfully. In [51] the P300 served to select the object on the screen. Then the object image will be scanned by a camera for the later grasping task. In [24], Xiaorong Gao proposed a BCI robotic arm system using SSVEP to complete the entire move-grasp-lift task successfully.

However, the applications of SSVEP, MI, and P300 potentials need constant human attention, which will cause user fatigue, and usually requires training phases for users to learn how to modulate thoughts appropriately. The main limitation of SSVEP and MI is the mismatch between mental commands and robot actions. For example, when using the MI paradigm, the user might need to map the mental command of moving left feet to a robot stopping. Similarly, in SSVEP, the user must memorize a mapping between flashing symbols and robot actions. The limited mapping to commands makes them incompatible with many HRI applications where people and autonomous robots engage in different tasks.

A more desirable approach would be one that utilizes a naturally occurring brain signal, thereby without interfering with the observer's other tasks and not requiring extensive training or active thought modulation by the person. The BCI paradigm based on error-related potentials (ErrPs), a cognitive phenomenon derived from EEG signals, is one such approach. With this mind, a unique feature of ErrP is that it would be involuntarily evoked when the human perceives unexpected errors in the environment. The user does not need to memorize an additional mapping between mental commands and robot actions. This permits detection and management of errors in any form and at any time during the interaction, promoting closed-loop passive BCIs as an efficient and seamless tool for online monitoring and evaluating of a robot's actions for users [5]. The ErrP phenomenon was first reported in choice-reaction tasks [41]. After the participant became aware of an erroneous response made by herself, a negative potential of approximately 80 ms and a sustained positivity in the time interval between 200 and 500 ms were observed [42, 55]. It was later found that ErrP was also evoked 250 ms after the user observed an unexpected event in the external environment [174].

Many previous studies have employed ErrPs for the correction of erroneous machine behaviours [36, 37, 48, 83, 98, 112, 148, 161, 194]. In particular, ErrP used as implicit input in a close-loop for real-time feedback on robot errors in real-world environments has garnered more attention in recent years. Salazar et al. [148] proposed a closed-loop system that used ErrP as an implicit input to guide the robotic arm in a binary bin-sorting task. Kim et al. [98] used ErrP as implicit reward for a real robot to learn the mapping between human gestures and actions. Stefan K. Ehrlich et al. [36] demonstrated the usability of ErrPs as feedback signals from the human for real-time mediating co-adaptation in human-robot interaction. Lopes-Dias et al. [112] showed the feasibility of online asynchronous decoding of ErrP and used it as feedback to control a robotic arm towards a target once the robot was halted at an unexpected moment. These experiments show that ErrPs decode human intention and be used as close-loop feedback towards a system during the interaction between humans and a device.

1.2 Problem definition

However, the ErrP needs to be precisely time-locked to a stimulus, such as the onset of a sudden robot movement, meaning that it is difficult to integrate the ErrP paradigm into a scenario that requires continuous user feedback [37, 111, 112, 148]. For example, it is difficult to use an ErrP-based BCI to stop a robot that continuously moves forward because the motion is continuous, and there is no clear event from the robot itself that an ErrP signal can be locked to. In [148], the ErrP was elicited by the robot's discrete movement actions. In [83], the robot arm's stop-action was used as the stimulus. Meanwhile, in [32], the sudden trajectory change functioned as an ErrP stimulus. The BCI system in which ErrP is elicited and used to control devices whose actions occur in a discrete way is well established. One potential breakthrough of continuous ErrP detection is more convenient for the users as interacting with the robot's performance during a continuous HRI task. So an important fundamental question is whether and to what extent the EEG-based ErrP obtained in a continuous HRI task is observable and decodable.

On the other hand, in most research, ErrP was used as either an implicit control signal to device [112, 148] or reward signal in reinforcement learning [3, 153, 187]. One breakthrough is to develop ErrP as an input feature in the human-robot collaboration, which allows human intervention in the control loop of the autonomous robot to achieve human-robot common goals. The shared control paradigm is such a method that allows interaction between humans and agents to achieve a goal. Traditionally, the user needs to give explicit input, such as keyboard and mouse commands [16, 143, 172], during interactions in most shared control tasks. The use of shared-control schemes opens the door to a new use of error-related potentials as a complementary signal for the BCI system.

However, due to the low decodability of ErrP, a direct mapping of partial observations to action is not sufficient for the desired behavior. Consequently, another research question concerns how to build an ErrP-based shared autonomy system in the human-robot collaboration that is robust to ErrP uncertainty.

1.3 Research aim and objective

To answer the research question about whether and to what extent the EEG-based ErrP is observable and decodable obtained in a continuous HRI task, this research proposes a novel ErrP-based interaction paradigm that allows continuous interactions between humans and the robot. To answer the research question of how to utilize ErrP in the human-robot collaboration that is robust to ErrP uncertainty, this research proposes a shared autonomy framework via deep recurrent learning. Under the research scope in this thesis, ErrP-based BCI for human-robot intelligent systems, the main and corresponding aims and objectives articulated below.

Aim 1: Explore the feasibility of using ErrP in a continuous human-robot interaction task.

Objective 1: This paper proposed a novel robotic design for continuous implicit robotic control using ErrP-based passive BCI.

Aim 2: Explore the feasibility of building a shared control system with ErrP.

Objective 2: This paper proposed a novel shared autonomy system incorporating ErrP information from the user as valuable observations for the robot. Our method jointly embeds the robot's observation of the environment with ErrP information from the user by concatenating them as a neural network input.

Aim 3: Explore the feasibility of the shared autonomy with uncertainty ErrP feedback using deep recurrent reinforcement learning.

Objective 3: We formulate the shared autonomy with uncertainty ErrP feedback as Partially Observed Markov Decision Process (POMDP). In this research, the Recurrent Neural Networks (RNN) which relies on actor-critic method was trained with simulated ErrP.

The following paragraphs introduce the approaches devised in this research that correspond to the research aims and objectives.

1.3.1 Continuous implicit Robot Control using Error-related Potentials

This research proposed a novel robotic design for continuous implicit robotic control using ErrP-based passive BCI. The key idea was to grant the robot an additional communication channel that enabled the autonomous robot to continuously signal its intended destination, from which the observer could evaluate the robot's intentions at any time and intervene early if necessary. We mounted an external liquid-crystal display (LCD) on the autonomous robot, which constantly communicated the intended destination via a flashing arrow. The LCD serves as an additional visual communication channel and allows a person to continuously evaluate the robot's intentions and intervene earlier, if necessary, before the robot commits an error. This novel implicit robotic control paradigm preserves the benefits of ErrP-based passive BCIs while permitting users to intervene earlier before the robot commits an error. We evaluated the proposed ErrP-based BCI system via an experiment where an observer controls a Robot Operating System (ROS) standard ground robot to perform a binary target-reaching task.

1.3.2 BCI-based shared control for human-robot interaction

Shared autonomy combines user input with automated assistance. Various methods have been used in shared autonomy. Research [56, 57, 152] proposed a flexible self-paced BCI by switching between automatic and subject control methods. While the switch model is efficient, there is only one control command at one time. This kind of method could not simultaneously take advantage of human inputs and robot autonomy. The weight-based shared control scheme that allocated adopted weights between human and autonomous robots is a popular method. In [162], the weight of mental command can be adjusted when performance is unreasonable in order to reduce the uncertain EEG pattern recognition. This approach [162] balances human preference and the autonomous controller. However, the autonomous command was generated based on its own information without taking advantage of human preference. Recently, deep reinforcement learning techniques have been used by blending user inputs and agent's observation [143]. In [143], they incorporate information from the user as helpful information for the agent by concatenating the user input and agent's observation.

This research takes this shared autonomy method [143] as a reference. We incorporate ErrP information from the user as useful observations for the agent. Our method jointly embeds the agent's observation of the environment with ErrP information from the user by simply concatenating them.

1.3.3 Shared autonomy via deep recurrent reinforcement learning

The use of shared-control schemes opens the door to a new use of error-related potentials as a complementary signal for the BCI system. DRL is a popular method used in shared autonomy [45, 88, 171]. By leveraging DRL, the performance of shared autonomy has been significantly improved [171]. This paper [88] utilized model-free deep Q-networks (DQN) to follow the user's actions closely enough and modifies the user's actions when they are suboptimal. This paper [45] uses DRL for haptic shared control to complete the task with the least amount of time.

However, DQN's performance declines when given imperfect inputs. Prior work [88] has formalized shared autonomy by modeling the system's task as Observed Markov Decision Process (POMDP) with the uncertainty of the user's intent. This paper [88] use QMDP [109] to solve the problem. The system models both the user input and robot action to map an assistance action. This paper [68] demonstrated that DRQN, which is a combination of Long Short Term Memory [71] and a Deep Q-Network [121], is capable of handling partial observability.

In this research, due to the low decodability of ErrP, a direct mapping of partial observations to action is not sufficient for optimal behavior. To describe the uncertainty about ErrP, we consider this as a Partially POMDP. We use the actor-critic method which relies on Recurrent Neural Networks (RNN) to deal with POMDP. RNN is an approach to stacking the memory history and robust to partial observation [68].

1.4 Structure of this Dissertation

Chapter 2 starts with a comprehensive literature review about brain-computer interface-based human-robot interaction, shared control paradigm in human-robot interaction, deep reinforcement learning used in shared control.

Next, Chapter 3 of this thesis will explain the different experimental designs used for this research project. This chapter will explain the experiment design. Also describe the different processes for data analysis and the methods used for DRL-based shared control are explained.

Chapter 4 will focus on the results of the feasibility of ErrP-based implicit robot control during continuous human-robot interactions.

Chapter 5 will focus on the training and test results of the ErrP-shared control method using DRL with simulated ErrP.

Chapter 6 will focus on the shared control model performance with real EEG data derived from real human participants.

Chapter 7 concludes the thesis and sets out the future direction in researching shared control model with intermittent interface

Chapter 2

Literature review

2.1 Brain-Computer Interface

EEG is one of the most commonly used approaches in BCIs. An EEG-based BCI allows humans to communicate with external devices with neural activities. The noninvasive BCI with electrodes placed on the scalp is widely used due to its convenience and the fact that it is harmless to the user. The brain potentials recorded by noninvasive electrodes are transformed into control commands to external devices.

There are different paradigms of EEG-based BCIs, such as P300 potentials [9, 164], SSVEP [43, 124], MI [44, 177], and ErrP [48, 51]. The P300 potentials are a positive peak at the latency of around 300ms of brain signal waves after stimuli. Stimuli with different frequencies visually evoked the SSVEP. The P300 potentials and SSVEP are based on external stimuli, whereas the MI is independent of external stimuli. The user needs movement imagination of certain limbs to generate the command.

As shown in Figure 2.1, the critical steps in the design of BCI intelligent systems based on EEG are to collect, preprocess and analyze EEG signals to identify the brain

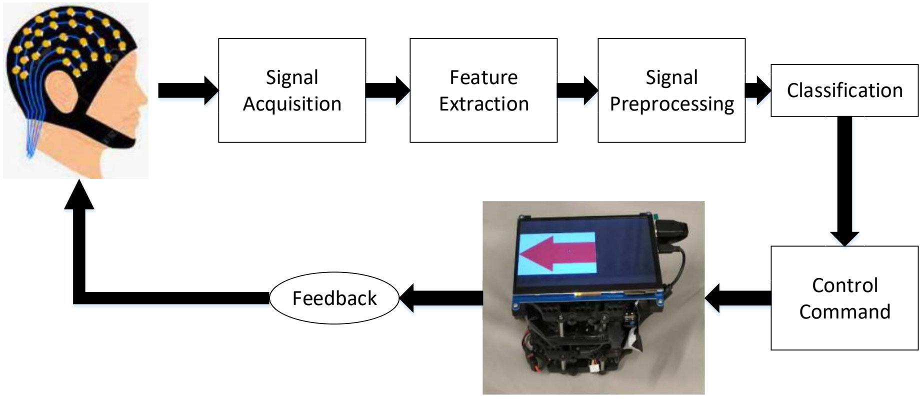


Figure 2.1: Critical steps of BCI system.

activity of the user. The different EEG paradigms can be decoded into control commands via feature extraction and classification algorithms. We review signals acquisition, preprocessing, feature extraction, and classification, especially on ErrP-based BCIs in the subsection below.

2.1.1 Signal Acquisition

The noninvasive EEG was recorded with electrodes put on the scalp's surface. The electrodes' locations are the standard of the 10-20 international system, which means that the actual distances between adjacent electrodes are 10% or 20% of the total front-back or right-left distance of the skull. Generally, the ErrP signals are recorded at the frontal to the central area locations with the ground and reference placed on the forehead and the right or left earlobe. In those papers [80, 85], the ground electrode was placed on the forehead and the reference channel on the left earlobe. The sixteen active electrodes were located at Fz, FC3, FC1, FCz, FC2, FC4, C3, C1, Cz, C2, C4, CP3, CP1, CPz, CP2, and CP4 (according to the 10/10 international system) during the ErrP experiment.

2.1.2 Preprocessing

Signal preprocessing serves to improve the signal-to-noise ratio. Filtering is widely used to remove artifacts, including low-pass, high pass, and band-pass filtering. In [20], it demonstrated that the error-related cognitive brain signals were mainly in the theta ([4-7]Hz) frequency range. In [80], the data was notch filtered at 50Hz, and the band-pass filtered at [1 10]Hz during the ErrP experiment. In [192], the signals were filtered with the band-pass frequency of 0.1 Hz and 15 Hz. Independent component analysis (ICA) is another frequently used method to eliminate artifacts of EEG signals.

2.1.3 Feature extraction

A feature represents a specific property of EEG signals. Feature extraction is meant to distinguish between different classes of EEG signals. Feature extractions can be divided into the time-domain feature, which is typically temporal variations amplitudes of time-locked potentials to a given/event/stimulus, and frequency-domain feature, which is typically frequency power spectra of EEG signals [8]. Normally, the ErrP and P300-based BCIs use temporal features, in which the MI- and SSVEP-based BCIs were based on band power feature [114].

For ErrP paradigm time-samples from a pre-defined window (usually between 200 and 600 ms) were used as features [10, 23, 30, 102]. The averaged time sample in a certain slide was usually utilized to reduce feature length [192]. In some cases, the automatic feature selection algorithms over large feature space, e.g. t-statistic, Fisher score served to quantify discriminant features [30, 59, 77]. These studies reported similar features that were selected but at better to capture subject-dependent variations of the feature. This paper [67] uses the inverse projection method as a spatial filtering technique that increases the signal-to-noise ratio of the discriminant features. The offline and online classification results demonstrated the validity of the approach. Alternatively, spatio-temporal filters and singular value decomposition approaches have been proposed recently.

The deep learning method was widely used in feature extraction for BCIs that allows automatic feature extraction without human intervention (citation: Deep Learning EEG Response Representation for Brain Computer Interface, A systematic deep learning model selection for p300-based brain-computer interfaces, EEGNet: a compact convolutional neural network for EEG-based brain‚Äìcomputer interfaces). In deep neural networks, each layer trains on a distinct set of features depending on the out of the previous layers (cite: Deep Learning-based Classification for Brain-Computer Interfaces).

A few studies explored frequency-domain features on ErrPs with general encouraging results [11, 135]. This paper [135] use theta power as a feature for three experiments, and they found that the performance degrades slightly using frequency features compared to those with the temporal feature. Another experiment [63] suggests the frequency-domain features may be less sensitive to temporal-domain features across individual trials.

2.1.4 Classification

The classification algorithms estimate the class of EEG data after feature extraction. Different classification methods can be chosen based on the BCI paradigms. In this section, we reviewed the classification methods that used in ErrP paradigm.

The most frequently used classification methods of ErrP are Linear discriminant analysis (LDA) [10, 107, 108, 138, 176] and support vector machines (SVM) [29, 181], as well as Gaussian classifiers [22, 47, 102]. In [80], single trial was classified as erroneous and correct by using LDA, and this particularly the case for online and real-time BCI. In those papers [83, 114, 178], the SVM has been used for classifying ErrP. This paper compared the performance between LDA and SVM classifiers, and they found little difference when training and testing on the same subject's data. Besides, the deep convolutional neural networks were used to decode EEG signals in a flanker task, where the average accuracy was significantly better than linear discriminant analysis classifier [179].

The classification can be divided into asynchronous classification and event-locked classification based on whether there is an event onset. Sometimes the event-locked classification is not applicable online due to missing information. In this case, asynchronous classification will be used to classify EEG signals. One paper [161] used both event-locked classification and asynchronous classification methods to classify EEG signal.

It is also divided into average-trial classification offline and single-trial based classification online. Typically, the average-trial classification was used for offline training. The single-trial-based classification detects the EEG features in real-time. An online BCI system is closed-loop and it was divided into two phases: 1) calibration phase used the pre-recorded data to train the classifier; and 2) use phase to recognize the EEG signal features and translate them into commands [113].

In some papers [48, 148, 194], the ErrP was classified online based on a single trial. Additionally, Millan [48] found that the ErrP potentials are relatively similar for all the participants. Therefore, it is reasonable to build one classification for all the subjects. In this case, it does not need to record ErrP to build the classifiers beforehand, which could simplify the training process.

2.2 Electroencephalogram (EEG)

The term EEG was first proposed and documented by Hans Berger in 1924 [78]. Since then, it has been widely used in noninvasive BCIs. However, EEG signals have a low signal-to-noise ratio (SNR). There are many aspects that can contaminate the signals with artifacts, such as scalp muscular activity, electrode movement, and environmental noise. Apart from the EEG signals being limited in spatial resolution events, they can provide exceptional temporal resolution. EEG has been distinguished with rhythms: delta (0.5-4 Hz), theta (4-8 Hz), alpha (8-13 Hz), beta (13-30), and gamma (above 30 Hz). EEG patterns can be influenced by intersubject differences, age, alertness, and mental disorders. The subject's behavior and age can both affect EEG rhythms. However, EEG technology is inexpensive, portable, and most important noninvasive and comfortable [70].

Noninvasive EEG was recorded by several electrodes placed on the scalp. The impedance of the interface between the electrodes and scalp should be less than 5KOhms (citation). The smaller the resistance, the higher the SNR. The electrodes' position is very important while recording EEG data. The standard 10-20 electrode system is an internationally recognized method for describing the electrode positions on the scalp [87, 140].

EEG of electrodes is measured as the potential difference relative to the reference electrode. The reference electrode is placed in a specific position, like the scalp central or earlobe position. Data recorded in each channel measured the potential difference. Thus, the common average reference montage is a common method to process EEG data to increase SNR.

The artifact rejection of EEG signals is a necessity. The artifact can emerge from body movement, nonstable electrode connection, external electromagnetic interference, etc. EEG is usually digitized by analog-digital conversion with resampling frequency for recording short-latency potentials.

Event-related potential (ERP). ERP-based BCI is one of the most popular BCI systems. ERPs are generated with ongoing EEG activities that are time-locked to a stimulus. The amplitudes with latencies usually describe ERPs. ERP is spontaneous EEG activity reacting to an external event. The ERP components at different latencies compared to the stimuli onset are different. The ERP waves consist of a series of position and negative components that are identified by latency and polarity. For example, the P300 component occurs at a peak around 300 ms after the stimuli onset. P300 sometimes also is called P3 designated by the serial order, meaning the third positive peak after the stimulus.

ERP can be evoked by visual, auditory, and somatosensory stimuli. ERP pattern is strongly related to the stimuli. The ERP component occurs at different latencies with different stimuli types and strengths. Averaging trials is a common approach for ERP analysis to reduce the artifact. The basis of averaging trials is under the assumption that the artifact is a random process and ERP is independent among trials. Time-domain identified by amplitudes and latency is a common form of ERP analysis. A second general ERP analysis method is frequency domain analysis by converting an amplitude latency series into a frequency function.

Error-related potentials (ErrP). ErrP is a goal-directed brain activity that originates from the frontal to the central cortex, mainly the anterior cingulate cortex [48]. ErrP is an event-related potential (ERP) elicited when wrong actions are perceived by the users and was first reported in choice reaction task [48, 51]. This is also called "response ErrP" [151]. There is a negativity of brain waves around 80 ms after erroneous response [48]. Later, ErrP was also found in situations where the user simply needs to monitor how well the system performs. ErrP activity is elicited when the user recognizes that the external system is incongruent with the users' intent, rather than when errors are made by themselves. This is also called observed ErrP. There will be a negativity around 250ms after the stimulus onset [24]. The significant difference of EEG activity between correct and incorrect trials suggests that ErrPs are particularly well-suited for the user to monitor the robot's feedback in a continuous HRI task.

2.3 BCI applications

There are many different BCI applications, such as brain-controlled wheelchairs or robots, word processors, and games. This section reviews the BCI applications to external devices, especially on brain-controlled robotics devices and ErrP-based BCI applications.

Many researchers have applied P300 to mobile robots [142]. In [116, 122, 193, 199], the SSVEP was used to control mobile robots. In [24], the SSVEP was used to control robot arm to grasp objects. In [48] J. del R. Mill used MI to move a cursor towards a target successfully. In [96], the MI was used online to control a lower-body exoskeleton in a closed loop. However, the main limitation of SSVEP and MI is the mismatch between the mental commands and the robot actions. For example, when using the MI paradigm, the user might need to map the mental command of moving left feet to a robot stopping. Similarly, in SSVEP, the user must memorize a mapping between flashing symbols and robot actions. The limited mapping to commands makes them incompatible with many HRI applications where human and autonomous robots engage in different tasks. A more desirable approach would be one that utilizes a naturally occurring brain signal without interfering with the observer's other tasks. It does not require extensive training or active thought modulation by a human.

The BCI paradigm based on error-related potentials (ErrPs), a cognitive phenomenon from EEG signals, is one such approach [46]. The ErrP occurs when the user notices an error made by the user or external agent. The observed ErrP is a component of event-related potentials. It is an electrical activity that is time-locked to external stimuli, such as tactile modality, visual, auditory, and spatial change. A lot of research has been done on the usage of ErrP in the control of robots. In [32], the author used the trajectory change as the stimuli of ErrP. In [194], the visual feedback was used to elicit ErrP. In [80, 81], the motion of the robot arm was used as the stimuli. In [35, 37], the author used LED on the screen as the event instead of the robot's actual motion, where the green color of LED represents correct and red means error. In [101] the LED triggered ErrP during the continuous movement of the robot. In [83], the action of the stop of the robot arm was used as the stimuli, and the zero of ErrP locked this moment. In [134], the author compared the sudden change of the expected trajectory of the cursor and the gradual deviation. However, it needs to note that the ErrP is goal-related brain activity. Those papers [103, 104] demonstrate that the ErrP in the fontal area is related to goal-directed feedback, and no ErrP in the frontal area was found related to the sudden visual stimuli when the feedback is correct. In [169], the author combined tactile and visual stimuli to enhance the single-trial detection of ErrP.

ErrP can also be used in hybird BCI. In [48], the computer can detect human intention via MI while ErrP was used for correction of there is misclassifications of MI to improve the quality of the brain-computer interaction.

2.3.1 Brain Controlled Robots

The EEG signals were translated into the command that operates a computer display or other devices in real-time. Many noninvasive EEG applications on robots have been reported. In [25], the user can freely control the wheelchair in real-time after training by using the motor imagery paradigm. Hua O. Wang [167] controlled the wheelchair to move left or right in the real world using only EEG signals. A wheelchair was controlled to move from avatar to avatar towards the end of a street within virtual reality via self-paced motor imagery [152]. The EEG signals were decoded in real-time to control an exoskeleton robot for assisting the stand-up movement [131]. In [185], MI was used to move the cursor to the target position. In [110], the MI was used to control the direction of a wheelchair, and the P300 potentials were used to control the velocity of the wheelchair through the graphical user interface. In [186], MI was used in upper limp rehabilitation for upper limb stroke patients. In [141], the P300 potentials were used to control the wheelchair in the automatic navigation model and direction model, and the eye-blink times were used for different pattern switches, such as fast stop, on/off P300 system, and model switch. In [117], the SSVEP signals are decoded to control an exoskeleton to move the patient's limbs. In another paper [65], SSVEP induced by one of the four LEDs with different frequencies served to control the direction of a robot to draw a fixed-sized line. Meanwhile, in [139], SSVEP was used in the navigation system. In [57], MI was used for the user to switch control modes: subject control (SC) and automatic control (AC). The user can control the moving direction during the SC model by MI. The author of [152] put forward the shared controller model in a wheelchair, which combined the EEG and autonomous robot.

Compared with other brain-controlled devices, brain-controlled mobile robots need to perform better (shorter classification time and higher accuracy) since they transport disabled people. Here, we reviewed the brain-controlled mobile robots-related vital techniques. Recently, brain-controlled mobile robots have received great attention since they can bring mobility back to disabled people who can not walk. Since 2004, Millan [119] successfully used EEG signals to control mobile robots. Many researchers have developed brain-controlled mobile robots systems.

2.3.2 Brain-Controlled Mobile Robots

Here we divided the brain-controlled mobile robots into two categories. One is that the EEG signal a direct control command to the robots. For example, Hua O. Wang [167] controlled the wheelchair to move left or right in the real world using only EEG signals. Andrzej Cichocki [25] confirms that they use EEG signals as the joystick that directly controls the wheelchair. The movement of robots depends on the EEG signals, which are slow and uncertain. Furthermore, users need to send the control command frequently but this can cause fatigue in users. The other one is that the EEG signals are used to control intelligent robots. The robot exhibits autonomous behavior to some extent. The EEG signals act as high-level decision-makers. The user just needs to send commands during the decision point, which can reduce fatigue so this is more reliable. Berdakh Abibullaev [132] used P300 potentials to control a mobile robot go in four directions (front, back, left, and right) as shown in Figure 8. Toshihisa Tanaka [193] developed a BCI platform based on SSVEP to control a mobile robot through the Internet remotely. In [2], a shared control system was built, where the SSVEP was used to provide high-level commands to an autonomous robot. Gerolf Vanacker [175] uses the MI to send the control command to the intelligent controller of a wheelchair, which is designed to filter out the potential erroneous command. In [116], the SSVEP was used to generate the target position, which was sent to the autonomous navigation systems. In [142], the P300 send the left and right intentions of the user to the wheelchair intelligent controller, which can generate the desired guiding paths. Meanwhile in [79], the P300 functioned to detect the target location, which was sent to the autonomous navigation system. However, little research about the usage of ErrP in mobile robots has been done till now.

2.3.3 ErrP-based human-robots interaction

Till now, ErrP was mostly used in simulated conditions rather than mobile robots in the real world. J. del R. Millan [47] implemented a cursor on the screen to represent the mobile robot. The user tries to move the cursor with a keyboard. The ErrP will be elicited if the movement of the cursor is not consistent with the user's intent. The purpose of this research was to explore whether ErrP could be elicited when the error is made by the interface during the recognition of the subject's intent. In [21], the user does not provide a control command to the simulation system, it simply monitors the agent's performance. The ErrP can also be observed if the cursor moves in the wrong direction. Seldom has employed ErrP in real-world robots. However, the key important thing is whether the simulation framework is potentially usable in a real robotic system, whether real robots can elicit the ErrP, and the feasibility for detecting it online and used to correct the robots' behaviors. Recently, more and more studies have focused on the ErrP in the real world. For examples, in [35–37], the ErrP was elicited by a humanoid robot. In [81, 148],the ErrP was triggered by the action of the robotic arm.

The application of observed ErrP has been reported in several studies [21, 32, 126, 151, 170].They show that the ErrPs can be used to decode human intention during the interaction between humans and a device. However, the evocation of ErrP needs to be precisely time-locked to external stimulus. In [32], the sudden trajectory change was used as ErrP stimuli. In [170], ErrP was elicited by the robot's discrete motion actions. In [126], the robot arm's stop-action served as the stimuli. The BCI system that ErrP was elicited and used to control devices whose actions occur discretely is well established. One potential breakthrough of continuous ErrP detection would provide the users with more convenience when interfacing with the robot's performance during a continuous HRI task.

The ErrP mentioned above was mainly based on the average trial, which can be easily observed. The averaging ErrP would be clearly observed if the potentials were deterministic signals superimposed in a linearly additive manner to the independent noise [177]. If the additive model is valid, averaging is efficient enough to recover the signal. Yet, when the ErrP was used as the control command of robots in real-time, single epoch identification is the key to the subject's feedback that can simultaneously send the command to robots. The classification of ErrP online, based on single-epoch classification, is more difficult than offline. Each experiment trial usually changes within an individual and the EEG signals are too noisy; the potentials are too small in amplitude to detect. Since 1977, Jacques J. Vidal [177] successfully identified the single epoch in real-time. Many researchers have attempted to classify EEG in real-time.

Millan [48] used the online classification of ErrP, and it improved the correct rate significantly during the interaction of the experiment where users employed MI to move the cursor to the target position. The recognition of ErrP in real-time was proved but the experiment was conducted in a simulation scenario. Javier Minguez [83] detected the ErrP that elicited while monitoring a five-degree of freedom robotic arm performing correct/incorrect reaching tasks in real-time. In [194], the author conducted a closed-loop experiment with real-time decoding of the ErrP in a real car. They found that the decoding of ErrP offline and online has similar performance. However, the ErrP was not used to control the vehicle in a closed-loop system in real-time directly. In [148], the ErrP was decoded in real-time and used in the closed-loop to correct the behavior of the robot. Elsewhere, in [148], the concept of primary ErrP and secondary ErrP that can greatly improve the classification performance was proposed. The amplitude of secondary ErrP is larger than the primary ErrP. However, the secondary ErrP was not used online as the user doesn't know the exact time of the robot's next behavior due to the hardware (robot's instability in free space). Table 2.1 summarised the BCI pardigms and controller device.

Controller device BCI paradigm	Simulated agent	Mobile robot	Robot arm	others
P300		[142], [132], [79]		[141], [142]
SSVEP		[116], [122], [193], [199], [139], [116]	[24], [65]	[117]
MI	[48], [185]	[167]	[96], [186]	[57], [25], [167], [152], [110], [175]
ErrP	[32], [80], [81], [134], [47], [21]		[46], [83], [81], [148], [126]	[35], [37], [101], [35], [36], [37], [194]

Table 2.1: BCI paradigms and controller devices.

2.3.4 Degree of freedom of BCI applications

This section reviewed the degree of freedom in the BCI paradigm. Understanding the degree of freedom of a neural command and the controlled device is fundamental. The control signals should be capable of actuating the device's degree of freedom. Ideally, the neural commands' degree of freedom should be full control of a scalar output, i.e., the ability to move in any direction and stop when needed. However, using noninvasive means to control a high degree of freedom is challenging, since BCI systems offer only a limited number of degrees of freedom. Even the most advanced BCIs have limited degrees of freedom and decode neural commands at the sub-prefect level. Direct control of multiple-degrees of freedom devices is difficult. Some research [191] needs to limit the degrees of freedom within a limited number. Some research focuses on how to increase the degree of freedom of BCIs. We reviewed the limitations concerning the degree of freedom of typical BCI paradigms, e.g., motor imagery (MI), SSVEP and P300, and the hybrid BCI, and how to increase the degrees of freedom.

The user's intended motion can be decoded using MI-based BCI. Most BCI systems rely on binary classification for brain activity, such as left- and right-hand imagery. The basic MI paradigm can decode 1-DoF while the user imagines left-hand and right-hand movements [64]. The robot cannot be controlled intuitively if the MI-based command cannot provide adequate DoFs. One paper [53] uses an adaptive user interface to offer multiple degrees of freedom control with a two-class MI-based BCI. In [128], motor imagery and speech imagery were combined with increasing the degree of freedom in an online feedback experiment. Elsewhere [118] decomposes three-dimensional space into two sequential low dimensional cursor control for multiple degrees of freedom control by reducing the number of degrees of freedom that the BCI needed to interpret. For basic motor-imagery BCI, the imagination of the right/left hand can be classified by a binary classifier. Multiple patterns [180] of MI can be extended to increase the degree of freedom. To summarize, MI method increases the mental task for the user to increase the degree of freedom. However, switching among multiple mental states will constantly increase the user's workload and make classification difficult. In order to increase the degrees of freedom, a multiclass classifier to map control commands and task [150]. However, the accuracy will be discounted with a multi-classifier. Indeed, it's much easier for the user to involve two mental states. Thus, the subsequent multi-step method based on binary classification may solve the low degree of freedom but it will decrease task efficiency.

SSVEP and P300: Although MI-based BCI provides more accurate results, SSVEP and P300 can offer more degrees of freedom based on the stimulus number [4]. SSVEP can control multi-DoFs devices with blinking flickers with different frequencies. In [24], SSVEP was used to control one-DoF pneumatic and a six-DoF robotic arm in real-time to realize a seven-DoF control. In [158], the SSVEP-based BCI system of multi-DoF manipulator was presented on the basis of virtual instruments. In [136], P300 was used to control a 9-DoF wheelchair through a graphical user interface (GUI). In [54] P300 was used to control a simulated two-degree of freedom robotic arm in a point-to-point real-time session. In this paper, P300 was used to control a 9-DoF system that combined a 7-DoF robotic arm and a 2-DoF wheelchair with a GUI. To summarize, the way for P300 and SSVEP to increase the degree of freedom is to increase the stimuli number.

Compared with single model BCI, the hybrid BCI system can achieve a high degree of freedom. Some papers [196] use hybrid BCI to increase the degree of freedom. This approach expanded the complexity of the system. A combination of SSVEP and P300 system was devised that can decode more than 100 commands code, which significantly increased the degree of freedom [188]. However, hybrid BCI requires more training procedures and mental workload for users to bear compared with a single BCI model.In [196], SSVEP was used to select the target object and use MI to control the four degrees of freedom of motion for robot grasping task. Other hybrid methods, such as eye-tracking [97] and fNIRS [95] were also used to increase the degree of freedom. Target selection and path planning are indirect ways to provide more freedom for the BCI system.

Using ErrP-based BCI to control many degrees of freedom all at once is difficult for a human, since ErrP-based BCIs only have two degrees of freedom. How to design a system that is not limited by the degree issue is an important question. Therefore, one of the critical objectives of this research is to design a BCI system with a low degree of freedom so that the robot with a higher degree of freedom can be controlled.

2.4 Human-Robot Interaction

HRI has been an extensive and diverse research field in recent years. HRI focus on developing a system where robots can interact with human users during a task. The collaboration in HRI obtains advantages for solving complicated problems. HRI has been widely used in different areas, such as search and rescue [86, 125], assistive robotics [105, 147], military and police [155], entertainment [15], space [50], and home and industry fields [52]. HRI is expected to make a positive difference in human life.

The HRI design is a common starting point for the HRI. Five key factors affect the interactions between humans and robots: level of autonomy, nature of information exchange, team structure, human-robot adaptation and learning, and shape of tasks [60]. Table 2.2 summarised the literature review that focus on different factors.

Key factors of human-robot interaction	Review source
Level of autonomy	[60], [69]
Information change	[75], [91], [13], [6], [5], [57]
Team structure	[145], [12]
Human robot adaptation	[58], [129], [146]

Table 2.2: Five key factors affect the interactions between humans and robots

The level of autonomy constitutes the critical aspect of designing an HRI task, which describes what level of the robot can act to accommodate environmental variations. Autonomy designing maps the inputs from the environment to the agent's actions. Based on what level the human and robot interact, the level of autonomy scaled from direct control to dynamic autonomy [60, 69]. On the direct control side, how to reduce the cognitive load of humans is one issue. Referring to the dynamic side, making robots interact naturally and efficiently with humans is another issue. A high level of autonomy system needs few interactions over a long period. Autonomy allows reducing and interaction frequency of the human user's workload in a task. This opens up technical challenges in dynamic environments and robot designs. The robots are not just a tool but interaction partners in HRI.

HRI is a field that understands and shares the interactions between humans and robots. Designing desirable interactions is an essential component of HRI. Interaction patterns are crucial. Human-Robot Interaction (HRI) requires communication between humans and robots. Communication can take several forms, like remote interaction and proximate interaction, according to whether the robot and human are in close proximity to each other [60, 75]. How to realize communication in HRI is a critical issue. Information exchange consists of two primary dimensions: the communications medium and format. In recent years, the media of HRI can be divided into visual displays [91], audios [13], haptics [6], and neural activities [5, 76] in recent years.

A typical HRI team consists of single human and a single robot [145]. As the technology develop, the team number is not restricted to two. More humans and robots, and software agents are coordinated in an HRI team [12].

Adaptation is a critical element of HRI, especially for the long-term interactions between humans and robots [58]. Also, learning is essential for both humans and robots so that they can interact with each other. The robots need to learn both offline design processes [129] and online interaction [146].

Task shaping describes how the task will and should be done. This includes goal-directed task analysis, cognitive analysis, and ethnographic studies [27, 39].

2.5 Shared autonomy in HRI

Human-robot interaction is a comprehensive research area where a human and robot interact with each other to complete collaborative tasks [28]. HRI research has arisen with the introduction of more advanced robotics. The HRI scenario can be divided into four aspects referring to this work [38].

- Independent: the human and the robot execute tasks independently with individual work pieces.

- Simultaneous: The human and the robot execute the same work pieces but without any dependency between them.

- Sequential: the human and the robot execute the same work piece with sequential process and time dependence between them.

- Supportive: the human and the robot work on the same process while executing the same work piece with time and task dependence between them.

There are many applications on independent and simultaneous tasks [144, 154, 182]. The map and human behaviors and robot actions are pre-defined, which is limited in the dynamic environment. In the sequential and supportive tasks, the robot needs to understand the human intention and optimize behavior under the interactions with humans. In [115], the human's next action was predicted by an anticipative interactions primitives model in an assembly task. In [74], the robot learned the human state and adaptively changed its policy in a handover task. The sequential and supportive approaches enable the robot behavior to be more flexible by collaborating with humans.

The shared autonomy (also known as shared control) belongs to the supportive method that has been widely used in HRIs. The shared control paradigm allows interaction between humans and agents that together achieve a goal. During shared control, the robot performs the search task with human supervision and feedback that were combined to achieve a task performance better than what individual partners alone can do. The robot can complete some simple tasks with a certain degree of autonomy. In [143] , the shared autonomy was used in assistive robotics with human input by operating a joystick. Shared autonomy was also used in industrial robots [40], semi-autonomous vehicles [149], and space robots [157].

2.5.1 BCI-based shared autonomy

Shared control is a widely used technology in human-robot interaction. This technique has been successfully used in the BCI system [18, 84, 173]. The BCIs offered new channels to allow shared control by integrating user intent directly from the ongoing brain activity, without the need to exploit any muscular control [82, 123, 190]. In [123], invasive BCI was used in a shared-control teleoperation framework. Elsewhere, in [190] motor imagery was used in shared control in a reach and grasp task, while [82] proposed a shared-control BCI using ErrP for a 2D reaching task.

There are two control components in the BCIs: direct control and shared control [8]. In a direct control model, the brain signals are decoded to directly control the robot's actions. The task is entirely dependent on the human user via BCI. The imperfect accuracy of the BCI system limits the intelligence of the system. The robot has more autonomy in shared control and is thus more robust to BCI uncertainty. Some paper [2, 79] used shared control in a hierarchical system, where the brain signal provides high-level commands via BCIs and the robot perform the low-level task, such as grasping, navigation, and manipulation. This shared autonomy system allows human user discrete feedback, which can reduce the [51] user's burden and fatigue. In [2], SSVEP was used to select the target, while the robot arm performed the specific grasp action. However, this shared control method subdivides tasks into separated modules for the human user and the robot. As well, the hierarchical shared control system was not robust to misclassification of EEG signals. Another shared control model was to mix human input and robot observation to generate a policy together. In this paper [143], the user information and agent's observation were concatenated as the input of neural networks to learn a shared autonomy.

The shared control scheme opens the door to a new use of ErrP as an alternative or complementary signal for the BCI system. Due to its nature, ErrP is particularly well suited as feedback or supervision signals during a shared control task. The user acts as a critic instead of voluntarily generating a command. The agent can exploit the evaluation of the current searching direction through user monitoring by changing the searching direction, which can improve searching efficiency. An agent can take advantage of humans' implicit brain signals when evaluating the agent's action. The human user does not need to explicitly send action commands, thus significantly reducing the burden on the human user.

In this research, ErrPs acts as a critic or supervisor and provides real-time evaluations or critiques to shape the behavior of the autonomous searching system interactively. The advantage of using ErrPs brain activity is that they do not require a person to perform demonstrations and only require an understanding of the task goal. In other words, the human user only needs to evaluate whether the current agent's action is good or not toward the final task goal. Another important issue is that the human user does not only provide feedback but also tend to provide guidance for future exploration by evaluating the robot's action serials. ErrPs provide direct access to the user's assessment of the robot operation without the ambiguities. The goal of the shared control system is to map the agent together with human evaluation to create a desired action.

2.6 Human-in-the-loop reinforcement learning

In this sextion, we reviewed human-in-the-loop RL. Some research learned a reward function from human feedback based on human preference [7, 26, 195]. The ErrP-based human-in-the-loop RL frameworks leverage ErrP as a positive or negative reward depending ErrP decoding result to accelerate training of autonomous agents that operate independently during the test [3, 153, 187]. ErrP was widely employed in reinforcement learning as the reward. In [82], inverse RL was used to infer the goal position based on ErrP in a virtual grid. In [153], ErrP is used in RL to update the reward to find a policy in route learning strategy. It is also used in RL to choose the correct target given several possible targets. In [81], ErrP served as the reward in reinforcement learning to train the intelligent controller of neuroprosthesis. The objective here was to improve the control policy. In [98], ErrP was used for the robot to learn the human gesture by using reinforcement learning based on the features originating from the Leap Motion and ErrP.

In [143], instead of being used as a reward, the agent receives human feedback as the agent's observation for model input in shared control. Human feedback can be a critique of an agent's current behavior rather than an alternative reward signal. Shared autonomy enables a semi-autonomous robot to interpret at test time. This is also the key difference between this method [143] and human-in-loop RL [3, 153, 187], which generally requires human evaluation during train but not testing. Table 2.3 summarised the usage in human-in-the-loop reinforcement learning, either used as rewards or input features.

ErrP usage in reinforcement learning	Review source
rewards	[3], [153], [187], [81], [98]
input features	[143]

Table 2.3: ErrP usage in reinforcement learning.

2.6.1 POMDP in Human-Robot Interaction

POMDP can deal with a sequential decision with uncertainties from human error feedback and sensing noises. POMDP formulate the problem that the state measurements are partial observations in sequential decisions. Quite recently, POMDP has emerged as a popular approach in human-robot collaboration tasks [14, 94, 130]. In the POMDP model, the state information is not fully observable but with partial observation or observation errors regarding the status of the environment, robots, and humans. POMDP can deal with a complex scenario where the agent's observation has a probability distribution. In this section, we review the application of DRL for solving POMDP problems.

In [93], human-robot collaboration was formulated as POMDP by characterizing the robot's information and human intention as the state space. In this paper [200], HRC was formulated as POMDP to learn the human model via Bayesian non-parametric learning to decide the human state. As shown in Figure 2.2, the environment, human, and robot states were incorporated with cross production. In another study [14], human intention is represented as an unobservable part of the state space in human-robot social interactions, which is modeled as POMDP. In [90] the human-aware motion was formalized as POMPD to address the sequential decision-making problems under the uncertainty of multimodal perception. As shown in Figure 2.3, the multimodal perception consisted of robot status, human behavior, and working condition and used for human intention prediction, autonomy sharing optimization, and robot motion policy learning. In [137], the human perception hand-off problem was formalized as the POMDP model when the driver perception module is uncertain about the environment in an HRI-based autonomous vehicle operation. In [61], a POMDP solver was devised to handle the infinite number of observations and large state spaces during the collaborative diaper change scenario. In [197], a supervisory control framework is designed to model uncertainties from the environment, humans, and robots based on the POMDP modeling. In [89], human behavior was formulated as a POMDP model, and the planning algorithm selected the optimal policy based on the history of human behavior. Meanwhile in [106], the observation model, dynamic machine model, and human model were combined into a framework and formulated as POMDP model for the human-in-the-loop system. In [73], human-computer interaction is formulated as a consequence of a POMDP, which is used to model human perception during interactions. In summary, POMDP does not assume full observability for the system states. The POMDP capability in representing

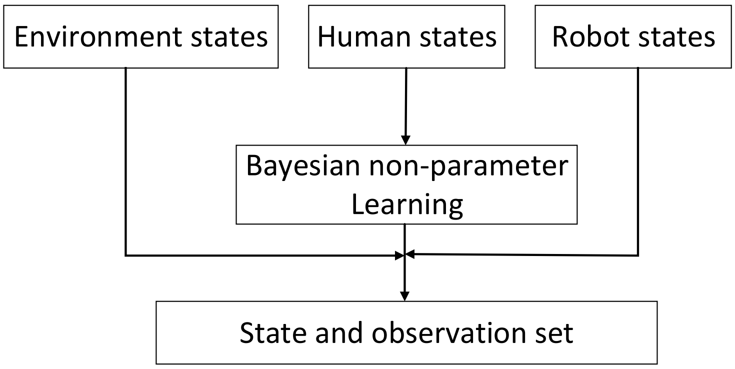


Figure 2.2: Bayesian non-parametric learning in POMDP.

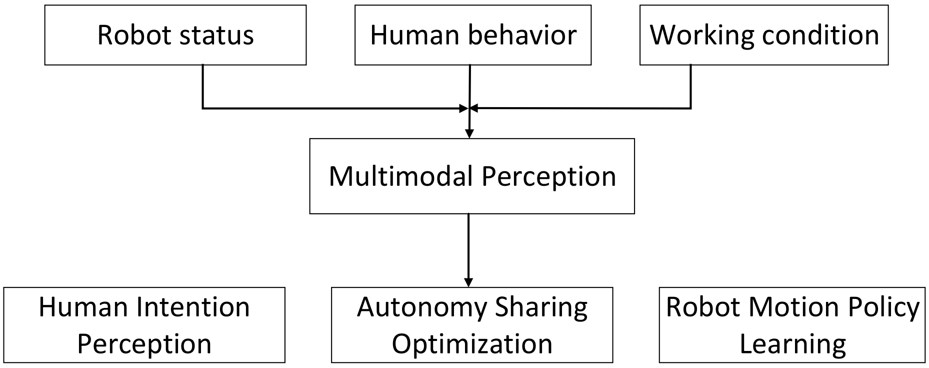

Robot status
Human behavior
Working condition
Multimodal Perception
Human Intention Perception
Autonomy Sharing Optimization
Robot Motion Policy Learning

Figure 2.3: multimodal perception model in POMDP.

uncertainties from different sources makes it a more suitable model in HRC applications.

Due to the low decodability of ErrP, the ErrP input variable is extremely noisy. For MDP, the best policy is invariably lost when there are imperfect observations. Similarly, human uncertainty feedback model the shared control as POMDP in our paper, where ErrP is decoded into binary value. The uncertainty of ErrP decoding can be represented by a Bernoulli Distribution with a probability p to observe the truth. Consequently, this can be considered as a Partially Observed Markov Decision Process with uncertain observation. POMDP makes it possible to make decision under conditions of uncertain input.

CHAPTER 3

MATERIALS, METHODS AND EXPERIMENT DESIGN

This chapter will explain the two different experiment designs utilized in this research. Each section will explain each experiment's materials and the methodologies followed by data analysis methods. Since the two experiments use same EEG recording and preprocessing, feature extraction, and classification procedure, we put them in a separate section. This chapter will also link how each experiment tries to address the different research objectives proposed in Chapter 1.

We put together of the two experiment in a same chapter, as the EEG recording and preprocessing, feature extraction, and classification were same in the two experiment. The first experiment design focus on ErrP-based implicit robot control during continuous human-robot interactions. The second experiment concentrates on ErrP-based shared autonomy via deep reinforcement learning. The second experiment was divided into a simulation experiment to explore the feasibility of the shared control autonomy; and a real experiment to validate the feasibility of the shared model with real human participants.

3.1 Experiment design of ErrP-based implicit robot control

To explore the feasibility of ErrP-based continuous interactions between humans and robots, this experiment proposed a novel robotic design for continuous implicit robotic control using ErrP-based passive BCI. The key idea was to grant the robot an additional communication channel that enabled the autonomous robot to continuously signal its intended destination, from which the observer could evaluate the robot's intentions at any time and intervene early if necessary. As an example, as shown in Figure 3.1 (a), we mounted an external liquid-crystal display (LCD) on the autonomous robot, which constantly communicated the intended destination via a flashing arrow. The LCD serves as an additional visual communication channel and allows the human to continuously evaluate the robot's intentions and intervene earlier, if necessary, before the robot commits an error. We hypothesized that when the robot intention deviates from the observer's expectations, e.g., the LCD flashes an arrow indicating a wrong target destination, an ErrP would be evoked before the robot commits an error. Once the ErrP was detected, an early correction command could be generated and updated the robotic movements, which was equivalent to the observer's implicit control of the robot behavior. This novel implicit robotic control paradigm preserves the benefits of ErrP-based passive BCIs while allowing users to intervene earlier before the robot commits an error.

We evaluated the proposed ErrP-based BCI system via an experiment where an observer controls a Robot Operating System (ROS) standard ground robot to perform a binary target-reaching task. As shown in Figure 3.1 (b), in each run, the ground robot moved from a starting point toward the t-junction, made either a left-hand or right-hand turn at the intersection, then reached the final destination at either side of the road. The ErrP signals evoked *before*, *during*, and *after* turning at the intersection were recorded and analyzed.

Participants Fourteen participants (average age 28.57 ± 3.11 years old, two females) with no previous BCI training participated in the experiment. The first ten participants participated only in offline sessions, while the last four participated in offline and online sessions. The data of one participant were excluded from further analysis, since that the participant was on medication during the experiment. All participants provided informed consent for the study, approved by the University of Technology Sydney (UTS) Human Research Ethics Committee (ETH19-3830). All participants had normal vision and did not report any known neurological or psychiatric diseases.

Hardware system We used Turtlebot3 [127], a ground-based mobile robot, in this study. TurtleBot is a ROS standard platform robot and integrated with 3D printing technology. An LCD was mounted on the robot's upper surface to communicate the robot's intention. The mobile robot was placed on a smooth cushion to reduce robot tremors while moving. A computer screen was put at the other end of the cushion facing the user.

Experimental Methodology As shown in Figure 3.2 (a), during the experiment, each participant wore an EEG cap and was seated before the cushion to evaluate the robot's intention during its continuous movement. At the start of each run, an arrow was shown on the computer screen to cue the human user's desired target destination (left or right at the end of the road) for the observer, which is also represented human intention. Then, the ground robot moved from a starting point toward an intersection with a linear velocity of 0.07 m/s, made either a left or right turn at the intersection with a rotation velocity of 1 rad/s. It then reached the final destination at either side of the road with a linear velocity of 0.07 m/s. Using the turning point at the intersection as a reference point, we could divide the experiment into three stages: before turning (Stage 1), during the turning (Stage 2), and after turning (Stage 3). As shown in Figure 3.2 (b), during all stages, the robot constantly shows the intended target destination (left or right) with

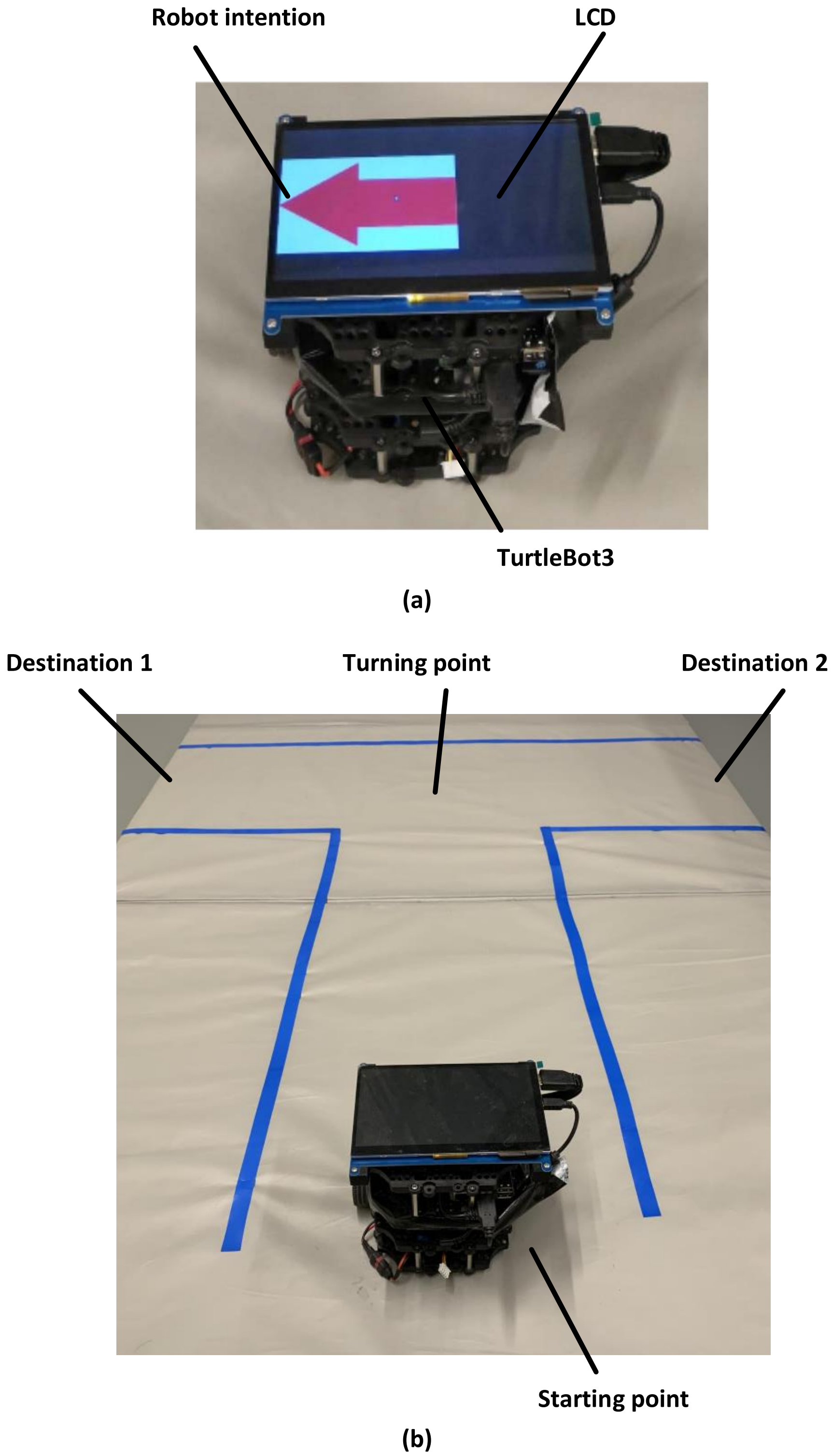


Figure 3.1: (a) LCD mounted on the ground robot. (b) The robot performs a binary target-reaching task.

either the LCD or robot turning movement. More specifically, as shown in Table 6.1, at stage 1, the robot signals its intended target destination with an arrow flash on the LCD, with the direction either left or right. The arrow flashed six times in a random sequence with the number of directions half left and half right. At stage 2, the robot signals its intended target destination with either left or right turning at the intersection. At stage 3, the robot signals its intended target destination with an arrow flashing three times on the LCD, with the direction either directing or deviating the target destination. At stage 1 and 3, the time duration between two arrow flashes is randomly chosen between 1500 ms and 2000 ms. There are 13 seconds at stage 1, 1.6 seconds at stage 2, and 6 seconds at stage 3. While the robot was moving, the observer was asked to fix his/her gaze at the robot. ErrP was expected to be detected at each stage if the robot intention was incongruent with human intention.

	Communication Channel	Implicit Robot Control
Stage 1	Arrow on the mounted LCD	Turning direction at the turning point
Stage 2	Turning motion of the robot	Correction of the turning direction
Stage 3	Arrow on the mounted LCD	Correction of the moving direction

Table 3.1: Robot communication channels and observer implicit controls command at different stages.

Offline sessions: No online classification was running while the robot signaled its intended target destination. After the robot finished stage 1 movement, it turned left 50% of the time, and it turned right the other 50% of the time at the intersection. Then it executed the stage 3 movement. The observer passively evaluated the robot's intention and was aware that their signals were not controlling the robot.

Online sessions: By evaluating the robot's intention, the observer's EEG signals are fed into the decoding model in real-time to correct the movement commands to the robot once the robot's intention deviates from the observer's expectation. Specifically, the robot intention at stage 2 and stage 3 can reflect the EEG classification result of stage 1 and stage2, respectively. As shown in Table 6.1, EEG signals at stage 1 were decoded to control the turning direction at the point of interaction. EEG signals at stage 2 were decoded to correct the turning direction at the point of interaction once a misclassification of EEG signals of stage 1 occurred. EEG signals at stage 3 were decoded to correct the moving direction once a misclassification of EEG signals of stage 2 occurred.

There are three offline sessions and one online session. Each session contains 40 runs. Each run contains 3 correct trials and 3 error trials at Stage 1, one trial at stage 2 (either correct or error), and three trials at stage 3 (either correct or error). Thus, each session contains 240 trials at stage 1, 40 trials at stage 2, and 120 trials at stage 3.

3.2 ErrP-based Shared Autonomy via Deep Recurrent Reinforcement Learning

This experiment mainly explore the feasibility of shared autonomy, which combines user input and agent observation to build a shared autonomy control system. As shown in Figure 3.3, we combine user input with simulated ErrP and robot observation as inputs of the deep model to learn a shared control strategy. We apply our method in a target search task. The goal location and agent start position are randomized at each episode in a constant static map through training and testing. Based on the RL model, the agent adapted its search direction to the expectations of the simulated human user feedback during the test. With a sequence of such adaptations, the agent can significantly improve efficiency in searching .

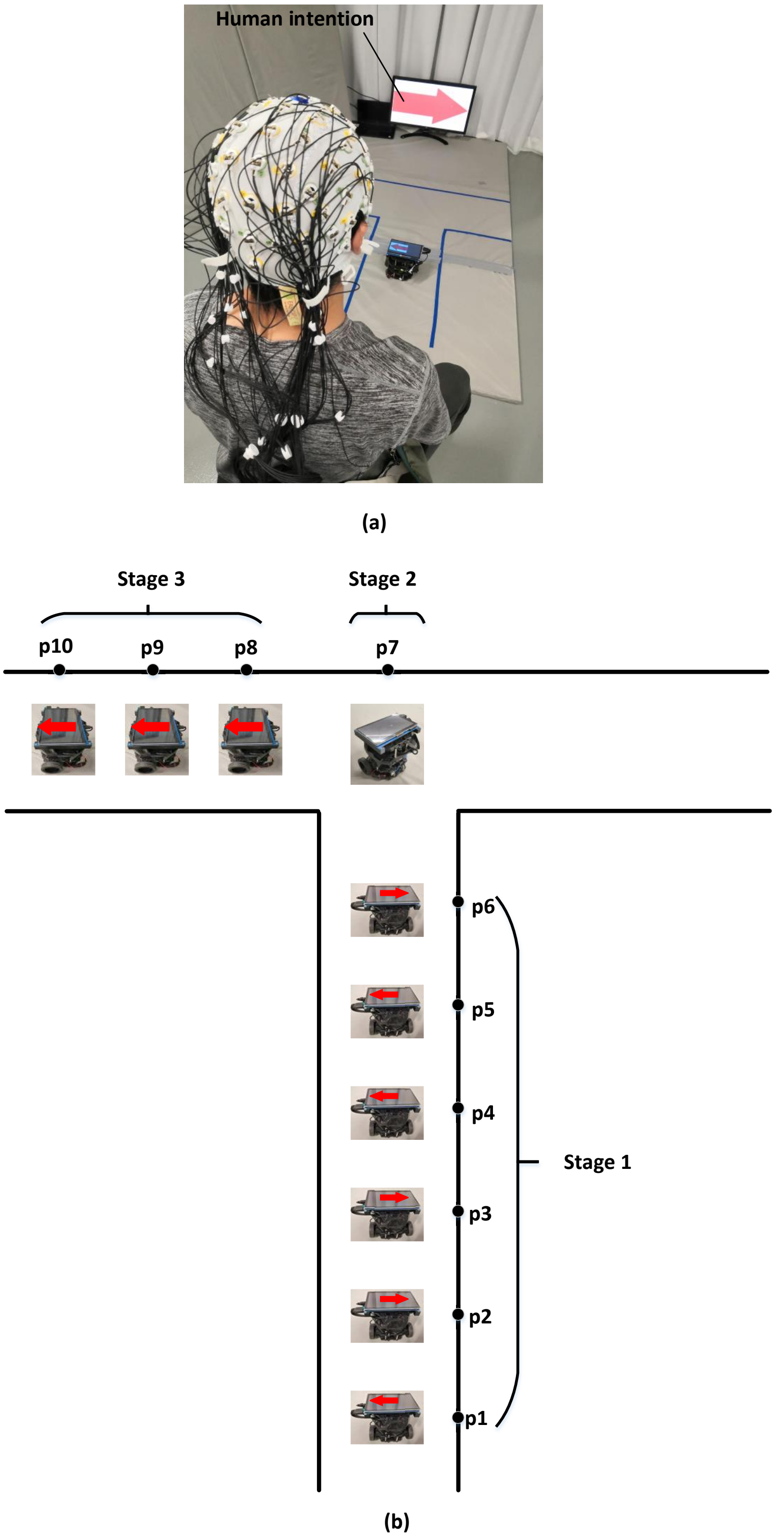


Figure 3.2: (a) Real scenario. (b) The robot signals its intentions ten times at three stages in one run. There were 6 times via LCD at the positions from p1 to p6 at Stage 1, one time at position p7 via turning movement at Stage 2, and three times at positions from p8 to p10 via LCD at Stage 3.

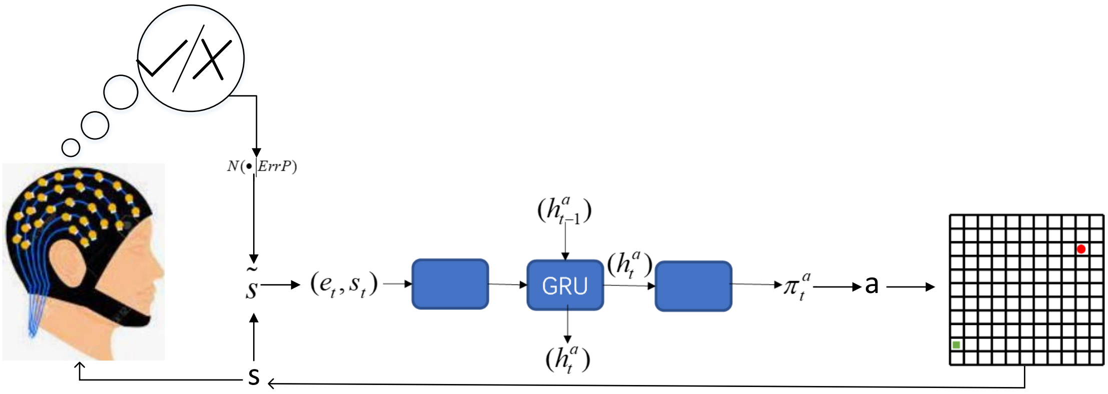


Figure 3.3: An overview of our method for ErrP-based real-time shared control autonomy and deep reinforcement learning. We evaluated our method in a target search task with real human participants. Here the red dot with an arrow is the agent, and the green square is the target

.

3.2.1 Background

3.2.1.1 Agents and environments RL

At each discrete time step t, an agent takes action and receives the observations provided by the environment. The environment also provides the next reward R_{t+1}, discount λ, and state S_{t+1}. The interaction is formalized as an MDP. A Markov decision process (MDP) is defined as tuple (S, A, T, λ, R) where states S, action A, transitions T: $S \times A \times S \to [0,1]$, reward function R: $S \times A \times S \to R$, and discount factor $\lambda \in [0, 1]$. In cases where the state is not fully observable, we can extend this definition to a partially-observable MDP (POMDP) in which there is an additional set of possible observation Ω and observation function O: $S \times \Omega \to [0,1]$. The expected future discount return of taking action a in state s with policyπ: $S \times A \to [0,1]$ is expressed by the state-action value function $Q^{\pi}(s,a)$, and the goal in RL is to learn a policy π^* that maximizes expected future discounted return. The agent's policy induces a value function V and an action value function Q. The advantage function is given by A=Q-V.

The REINFORCE algorithm [184] iteratively updates a given policy π^* in the direction of the optimal policy. The update direction is defined by the gradient:

$$g = E_T \left[\sum_{t=0}^{T-1} \nabla_{\theta\pi} \log \pi(u_t | s_t) G_t \right]$$

REINFORCE samples the gradient estimate by collecting trajectory samples and then computing the empirical returns G_t. This introduces inherent high variability of the gradients [62]. In actor-critic approaches [99, 156, 166], the policy is trained by following a gradient that depends on a critic, which usually estimates the a value function.

In particular, G_t is replaced by any expression equivalent to $Q(\tau^a, u^a) - b(s_t)$, where $b(s_t)$ is a baseline designed to reduce variance [183]. A common choice is to use value function V as baseline. The agent's critic estimates $Q(\tau^a, u^a)$ and follows a gradient based on the advantage:

$$A(\tau^a, u^a) = Q(\tau^a, u^a) - V(\tau^a)$$

Where

$$V(\tau^a) = \textstyle\sum_{u^a} \pi(u^a \left| \tau^a \right) Q(\tau^a, u^a)$$

$A(\tau^a, u^a)$ is called advantage function. This is called Advantage Actor-Critic (A2C) algorithm.

Another option is to replace G_t with temporal difference (TD) error $r + \lambda V(s+1) - V(s)$, which is unbiased estimation of advantage function A. In particular, the gradient must be estimated from trajectories sampled from the environment, and the value function must be estimated with function approximators or neural networks. Consequently, the bias and variance of the gradient estimate depend strongly on the exact choice of the estimator [99].

In this paper, we train critic fc on-policy to estimate Q value, using a TD(λ) [166] adapted for use of deep neural networks. TD(λ) uses a mixture of n-steps returns. In particular, the critic parameters θ are updated by minibatch gradient descent to minimize the following loss:

$$\delta(\theta^c) = (y^{(\lambda)} - f^c(\cdot t, \theta^c))^2$$

Where $y^{(\lambda)} = (1-\lambda)\sum_{n=1}^{\infty} \lambda^{n-1} G_t^{(n)}$, and the n-step return $G_t^{(n)}$ are calculated with bootstrapped values estimated by a target network [49, 121] with parameters copied periodically from θ^c.

We choose the Advantage Actor-Critic (A2C) algorithm [31, 166] to perform RL. At every time step, the policy architecture is fed with ErrP and agent local observation and step number and is tasked with estimating the Q-value function and policy at every point.

3.2.1.2 Partially Observable Markov Decision Processes

Markov Decision Process (MDP) works on the assumption that the agent knows the full observation of the environment. Put another way, the agent sensors the environment with limited observation or uncertain observation. The uncertain observation means the state signal will no longer be Markovian, violating a key assumption of most reinforcement-learning techniques [165]. A partially observable Markov decision process allows for the best decisions to be made under partial observability of the agent [92]. A partially observable Markov decision process is a tuple $\langle S, A, \Omega, T, O, R \rangle$in which S is a finite set of states, A is a finite set of actions, Ω is a finite set of observations, T is a transition function defined as T: $S \times A \times S \rightarrow [0,1]$, O is an observation function defined as O: $S \times A \times \Omega \rightarrow [0,1]$, and R is the reward function defined as R: $S \times A \times S \rightarrow R$.

The discrete set of observation $\Omega = \left\{o^1, ..., o^M\right\}$ represents the agent observation, which depends on the next state $s^{'}$, and sometimes conditioned on its action a. It can be drawn by the observation function O: $S \times A \times \Omega \rightarrow [0,1]$. The observing o probability in state $s^{'}$ after an action is $O\left(s^{'}, a, o\right)$. It required $O\left(s^{'}, a, o\right) \geq 0$ and $\sum_{o \in \Omega} O(s^{'}, a, o) = 1$. In this thesis, the discrete partial observation $\Omega = \{0, 1\}$, which represents the decoding result of ErrP. The probability is the Bernoulli distribution. If the p=0.7, it can be modeled as follows. $O\left(s^{'},a,o^{1}\right)=0.7$, $O\left(s^{'},a,o^{2}\right)=0.3$, or $O\left(s^{'},a,o^{2}\right)=0.7$, $O\left(s^{'},a,o^{1}\right)=0.3$. It can be argued the agent is equipped with a sensor that knows the experiment is correct in 70% of the cases.

3.2.1.3 ErrP-based human feedback: MDP or POMDP

While using simulated ErrP with 100% accuracy, the learning process can be formulated as MDP. However, the classification of ErrP collected from humans in a real environment is not perfect due to misclassification. ErrP uncertainty can be considered as the agent's imperfect sensors of the true state regarding the environment.

For instance, let us suppose the ErrP feedback as the robot camera to identify the arrow traffic sign, which is likely to make mistakes sometimes and report the wrong identity. The imperfect sensor prevents the agent from knowing the true stage of the environment. For example, if the arrow is pointing to the left but the camera sensed right. A partially observable Markov decision process allows for optimal decision when the environment is partially observed. In general, partial observability stems from two sources: (1) the agent can only sense the limited part of the environment, and (2) the sensor is uncertain with noise reading: observing the true state can result in different observations. The POMDP captures the partial observation in a probabilistic model, which relates possible observed states. In our paper, the ErrP uncertainty belongs to the second condition with the probabilistic model as Bernoulli Distribution. The probability of ErrP observation error is p_e.

3.2.2 Method and experiment design

3.2.2.1 Network architecture

The design of the network architecture was referred to in [49]. The actor consists of 64-bit gated recurrent units (GRUs) that use fully connected layers both to process the input and to produce the output values from the hidden states, h_t^a. Action probabilities are produced from the final layers, z, via a bounded softmax distribution that lower-bounds the probability of any given action by $\varepsilon/|U| : P(u) = (1-\varepsilon) soft\max(z)_u + \varepsilon/|U|$. We anneal ε linearly from 0.5 to 0.05 across 5500 training episodes and set it to 0 during test. The critic is a feedforward network with multiple ReLU layers combined with fully connected layers.

3.2.2.2 Task statement

We test our method on two environments. As shown in Figure 3.4, the target search environment is described by a grid map. The first environment is a grid map without obstacles while the second environment is a grid map with several obstacles. The map layout and the obstacles were fixed throughout training and testing.

The size of the grid we selected was 11 × 11. The location of the agent and target was simplified as the coordinate of the grid. The horizons of the agent were limited to its neighbored four positions, with 1 × 1 horizon to detect the target. The agent can only move North, South, West, and East in each time step and cannot move to the barriers or out of the grid. The task for the agent is to find a goal location within the map. The agent has a limited sensing range, which is assumed to be much smaller than the maze size. The detection or capture of the target will happen when the target is in the agent's sensing range. The goal location and agent start position are randomized (spawned) at each episode in a constant static map through training and testing. Once the goal was found, a new episode began. To encourage short trajectories, there is a step cost

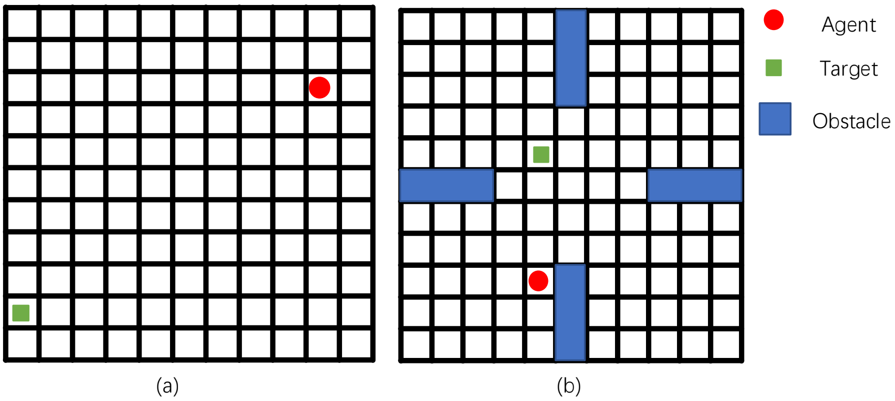


Figure 3.4: The environment without obstacles (a) and obstacles (b).

(penalty) in each time step. A typical sparse terminal reward and a step cost are provided to encourage the agent to reach the target position with minimal steps.

3.2.2.3 Network input features

The agent receives its own observation and human feedback. The agent observation includes current and four adjacent positions (x, y) and the neighbor's vision of four adjacent grids: whether the target is at the adjacent position and whether the agent has visited it. Apart from this, the step number, step cost, and last agent action are included. All features are normalized by their maximum values. Any information about the target position was not included in the input. The human feedback is with simulated ErrP. We incorporate information from the user as useful observation for the agent by simply concatenating the agent observation and human input.

Last action: The coupled action was implemented as input as well. The authors demonstrated [201] that the coupled action-observation pair would improve the performance, especially during the training process. It is true that ErrP is cued by the agent's last action. The action and observation are paired. In [201], actions are encoded via a fully connected layer to match observation vectors. However, in this thesis, the observation vector is a small size. We simply concatenate the action and observation vector.

Mark of visited grads: For a stationary target, an ideal search strategy is simply represented by a path that tries to cover the entire environment without revisiting any location. To encourage the agent to search unvisited location, the visited area was marked. This can be sensed by the agent within its vision (four adjacent grads around the agent's current position). The marker was used as input rather than a reward. We found that when the visited marker was used as input proved to better than as a reward. If visited, the reward was positive, otherwise it was negative. The poor performance of using as a reward because when given the sparse reward means the action that goes to the visited area was bad, and good when the agent goes to the non-visited area. However, it can be a good action to the visited area even though it is visited, since reaching to the target needs to go through this visited node. The reward explicitly encourages the agent to explore the non-visited area by rewarding the agent. Yet, considering a situation when the neighbor's area was visited by the agent, the optimal action was going to the non-visited action. Using the visited marker as input, we found that the model will learn an optimal strategy. The given reward was limited to the optimal policy. Reward for action that visiting unexplored areas yields slightly less desirable policy. Removing the reward and using it as input only results in a better policy. Many research papers use rewards to encourage exploration [120], but the reward needs to be carefully designed.

ErrP simulation: To eliminate the gap between simulated EEG and real EEG data collected from a person, we simplify the EEG data as a binary variable, which corresponds to the decoding output of the ErrP classifier. During model training, we use binary 0 and 1 to simulate the output of the ErrP binary classifier. To simulate ErrP during training, we calculate the shortest path towards the target position at each step. The shortest path [33] was computed with the information of the full map environment.

This is the same principle whereby the human user has a global view of the environment. If the current shortest path was longer than the last one, we consider the current step is a bad action and makes ErrP label noted as 1, otherwise 0.

3.3 Shared autonomy validation with real human participants

The main purpose of this experiment is to evaluate our shared control model during the test phase with real human participants. The model we used is pre-trained with simulated EEG. We want to validate the feasibility that the model trained with simulated EEG can be used in real human participants in the same task environment.

3.3.1 Experiment

Participants Seven participants (one female) took part in the experiment. All the participants were involved in both offline and online sessions. The data of three participants were excluded from further analysis, as the ErrP classification accuracy was lower than 70% during training using 10-fold cross-validation. All participants provided informed consent for the study, which was approved by the University of Technology Sydney (UTS) Human Research Ethics Committee (ETH19-3830). All participants had normal vision and did not report any known neurological or psychiatric diseases.

Interaction scenario design In order to evoke ErrP signals, the interaction scenario, especially the stimulus, needs to be carefully considered. The scenario design was referred to as Klaus's design [192]. As shown in Figure 3.5 (b), there is a gray grid with a red agent and a green target on a black background. The agent's start position and target were generated with their distance being bigger than 1 (the agent's observation ability). The agent moved from its current position to one of the four adjacent positions in each

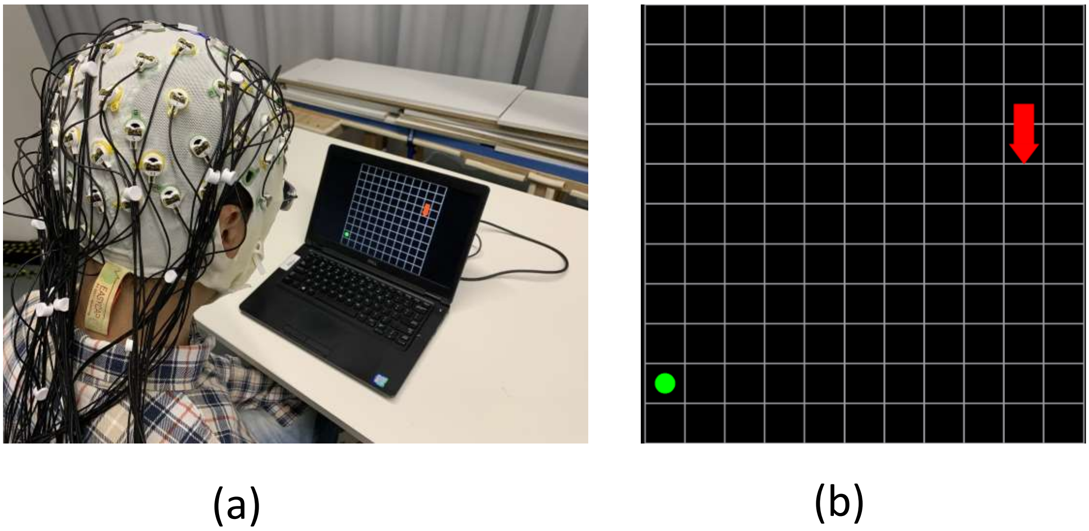


Figure 3.5: We evaluated our method in a target search task with real human participants (a). The red dot with an arrow is the agent , and the green square is the target (b)

.

step. A 1-s animation within the agent served as a countdown to draw the participants' attention. The agent would then instantaneously jump to the next position with an arrow directing to the position. This arrow remained visible for 1 s. Following that, the highlights disappeared, and the agent would remain at its new position for 1 s more before the next step.

Offline sessions: Before real-time control, As shown in Figure 3.5 (a), participants were asked to first perform five blocks of 120 trials on scenario 1 without obstacles. The agent's initial position and target position were randomly generated. If the target had not been reached the target position after 60 trials, a new run was started. The EEG data collected during these five blocks was used to calibrate the classifier. The agent's action with the movement with the distance close to the target position was labeled as "correct," and the movement far away from the target position was labeled as "error." The participants mentally judged whether the agent's action was correct or wrong during the experiment.

Online sessions: The online experimental paradigm involved two scenarios, an 11×11 grid without obstacles and an 11×11 grid with obstacles. In this section, we practically integrate real EEG data to validate the feasibility of the shared control model. The EEG signals are classified in real-time to be fed back to the shared control system. The ErrP classifier was trained with offline EEG data. We use two scenarios: one without obstacles and one with obstacles. To test the model's target searchability with different distance scape, we choose samples with an initial distance $d = |x_T - x_0| + |y_T - y_0|$ from 2 to 20 (maximum). There are in total 19 samples with random sequences for each scenario. The samples were pre-generated for all participants. The 19 samples were divided into two blocks, with the first block ten samples and the second block nine samples to let the participants rest between the two blocks. Consequently, there are four blocks for the two scenarios during the test. The 19 samples were randomly generated separately for the two scenarios. All the participants used the same group of pre-generated samples.

3.4 EEG processing and classification used for the two experiments

3.4.1 EEG recording and preprocessing

EEG signals were recorded from 64 locations according to the extended 10/20 system using a LiveAmp wireless EEG system with a sampling rate of 500 Hz. The reference channel was placed at the FCz channel position, and the ground channel was placed at the forehead position [1]. The signal was resampled to 256 Hz and filtered using a finite impulse response (FIR) bandpass filter with cut-off frequencies 1-50 Hz. Then, the common average reference served to reduce signal contamination.

3.4.2 Electrophysiological analysis

After preprocessing, we extracted correct and error trials according to the onset of the cued human intention. The correct and error trials averaged over all participants in the offline experiments were segmented into epochs by extracting time intervals of -300 to 800 ms relative to the onset of the visual stimuli. We expected to observe different ERPs in the participants evoked by correct and error trials. We choose Fz channel to examine the ErrP, which is located at the frontal area and is normally used to analyze ErrPs [23, 192]. As ErrP is a goal-directed brain activity that originates from the frontal to the central cortex, mainly the anterior cingulate cortex.

3.4.3 Feature extraction

Temporal features extracted from time-series data have been used in many ErrP activity studies [37, 148, 159]. It has been reported that the classification results of temporal features are better than those from spectral features in the context of decoding the ErrP [83]. Some research [133, 161] considered frequency domain features in their classification. In the context of single-trials classification of error-related potentials, temporal features extracted from time series have been shown to lead to high classification performance and mostly outperformed other types of features [35, 47, 81, 83]. Similarly, we obtained comparable results using only time-domain features. Alternatively, one could approach the detection of error-related potentials from a more generic perspective by taking into account frequency ranges that are not commonly considered. Temporal features were used for classification in this study. The averaged signal amplitude within a 30-ms-long window between 150 ms and 600 ms at each trial and channel was extracted. Thus, during the time window ranging from 150 ms to 600 ms, there will be 15 ((600-150)/30) time points for one channel. The classification between correct and error feedback was done from all 64 EEG electrodes. So the feature vector length is 64*15=960.

3.4.4 Classification

Classification of ErrP is challenging due to the high-dimensional, noise-contamination EEG data. To minimize overfitting effects, we used ten-fold cross-validation to train the classifier with 90% of the data, and the remaining 10% of the data was used for testing. Since the extracted features include redundant features, traditional linear discriminant analysis (LDA) had limited flexibility when the feature was complex. It is necessary to search for a subset from the available features that can improve classification performance . Besides, the reduced feature can also improve the decoding efficacy for real time implementation. Shrinkage and selection methods are commonly used methods for feature selection. We compared different classification methods, including shrinkage LDA [107], stepwise LDA (SWLDA) [34], support vector machine [29] , and elastic net [66]. The classification performance (overall accuracy (ACC) is reported.

1) Shrinkage LDA

A binary linear classifier can be characterized by a projection vector **w** and a bias term b referring to the separating hyperplane $\mathbf{wx} + b = 0$. The projection vector of LDA is defined as:

$$w = \boldsymbol{S}_w^{-1}(\boldsymbol{u}_a - \boldsymbol{u}_b) \tag{3.1}$$

Where $\boldsymbol{S}_w^{-1}$ is the covariance or within class variance, $\boldsymbol{u}_a$ and $\boldsymbol{u}_b$ is the mean value of class A and class B.

The empirical covariance is unbiased and contains good properties when the number of observations is greater than the dimensionality of variables. However, for high-dimensional data with only a few data trials, the estimation covariance may become imprecise because the covariance of matrix estimate is singular, and the inverted matrix is imprecise. This phenomenon leads to a systematic error: large eigenvalues of the original covariance matrix are estimated too large, and small eigenvalues are estimated too small [67]. This estimation error makes the performance of LDA in high-dimensional situations far from optimal. Shrinkage is a common method that compensates the systematic bias in the estimated covariance matrix by a regularized covariance matrix $\boldsymbol{S}_b$:

$$\boldsymbol{S}_b = (1-\lambda)\boldsymbol{S}_b + \lambda\boldsymbol{D} \tag{3.2}$$

Where D is a diagonal matrix taking the diagonal elements of $\boldsymbol{S}_b$. Thus, the parameter λ forces the extreme eigenvalues towards average [67].

2) SWLDA

In essence, the stepwise method identifies the time window and scalp channels along with the ERP epoch, which best differentiates between the groups of ERPs in the training data. The stepwise feature selection model is computed in a training set using CV. The selection model can then be used to classify a test set whose label is "unknown".

Stepwise extracts a subset of features by employing forward and backward regression models based on their statistical significance. This method begins with an initial model and then takes successive modification steps by adding or removing variables. In each forward step, the F-statistics and p-value of the variable are computed. If the p-value <0.1, the variable is added to the model. After each variable is entered into the model, a backward regression analysis is executed to remove the variables if the p-value >0.15. Here, the maximum feature number was restricted to 60, considering the limited number of samples in this study. On the other hand, stepwise can also be seen as a shrinkage method by reducing the size of feature by shrinking the coefficient estimates towards zero. If the coefficient shrinks to exactly zero, the corresponding variables are removed from of the model.

3) Support vector machine (SVM)

A linear SVM is designed to determine the hyperplane that maximizes the separating margin between the two classes in binary classification. Given a training set of instance-label pairs (x_i , y_i), where $x_i \in Rn$ and $y_i \in \{1,-1\}$, the SVM [19] solve the following optimization problem:

Subject to

$$y_i(\boldsymbol{w}^T \Theta(\boldsymbol{x}_i + b) \geq 1 - \zeta_i, \zeta_i \geq 0 \tag{3.3}$$

Where y_i is the class label, ζ_i is the distance from the misclassified points to the margin.

4) Elastic net

The elastic net algorithm is based on the alternating direction method of multipliers (ADMM) [66]. For a given estimation of lamda, a nonnegative parameter, and alpha strictly between 0 and 1, elastic solves the problem:

$$\min_{\beta_0,\beta}(\frac{1}{2N}\sum_{i=1}^{n}(y_i - \beta_0 - x_i^T\beta)^2 + \lambda P_a(\beta)) \tag{3.4}$$

where

$$P_a(\beta) = \frac{1-a}{2}\|\beta\|_2^2 + \alpha\|\beta\|_1 = \sum_{i=1}^{p}(\frac{1-a}{2}\beta_j^2 + \alpha|\beta_j|) \tag{3.5}$$

Where, N is the number of observations, y_i denotes the response at observation i, x_i stands for data (a vector of length p at observation i), λ represents a nonnegative regularization parameter corresponding to one value of lambda, the parameters β_0 and β are a scalar and a vector of length p, respectively.

CHAPTER 4

EVALUATION OF IMPLICIT ROBOT CONTROL USING ERRP

This chapter presents the experiment overview, results analysis, and discussion based on the experiment design explained in Chapter 3, Section 3.1. The results include ERP analysis over different interaction sequences and offline and online classification performance. Discussion includes ErrP observability and decodability at different interaction sequence.

4.1 Experiment overview

To explore the feasibility of ErrP-based continuous interactions between humans and robots, this research proposed a novel robotic design for continuous implicit robotic control using ErrP-based passive BCI. The key idea was to grant the robot an additional communication channel that enabled the autonomous robot to continuously signal its intended destination, from which the observer could evaluate the robot's intentions at any time and intervene early if necessary. We mounted an external liquid-crystal display (LCD) on the autonomous robot, which constantly communicated the intended destination via a flashing arrow. The LCD serves as an additional visual communication channel and allows the human to continuously evaluate the robot's intentions and intervene earlier, if necessary, before the robot commits an error. This novel implicit robotic control paradigm preserves the benefits of ErrP-based passive BCIs while allowing users to intervene earlier before the robot commits an error.

We evaluated the proposed ErrP-based BCI system via an experiment where an observer controls a Robot Operating System (ROS) standard ground robot to perform a binary target-reaching task. As shown in Figure 4.1, in each run, the ground robot moved from a starting point toward the t-junction, made either a left or right turn at the intersection, then reached the final destination at either side of the road. The ErrP signals evoked *before*, *during*, and *after* turning at the intersection were recorded and analyzed. 14 participants with no previous BCI training were recruited.

We hypothesized that when the robot intention deviates from the observer's expectations, e.g., the LCD flashes an arrow indicating a wrong target destination, an ErrP would be evoked before the robot commits an error. Once the ErrP was detected, an early correction command could be generated and updated the robotic movements, which was equivalent to the observer's implicit control of the robot behavior. This novel implicit robotic control paradigm preserves the benefits of ErrP-based passive BCIs while allowing users to intervene earlier before the robot commits an error.

4.2 Results analysis

4.2.1 ERP analysis

4.2.1.1 Event-related potentials for the averaged sequence

Here, we analyze the ERPs of the three stages separately across all the participants.

(a) The ERP of stage 1: As shown in Figure 4.2 (a), we observed clear ErrP activity

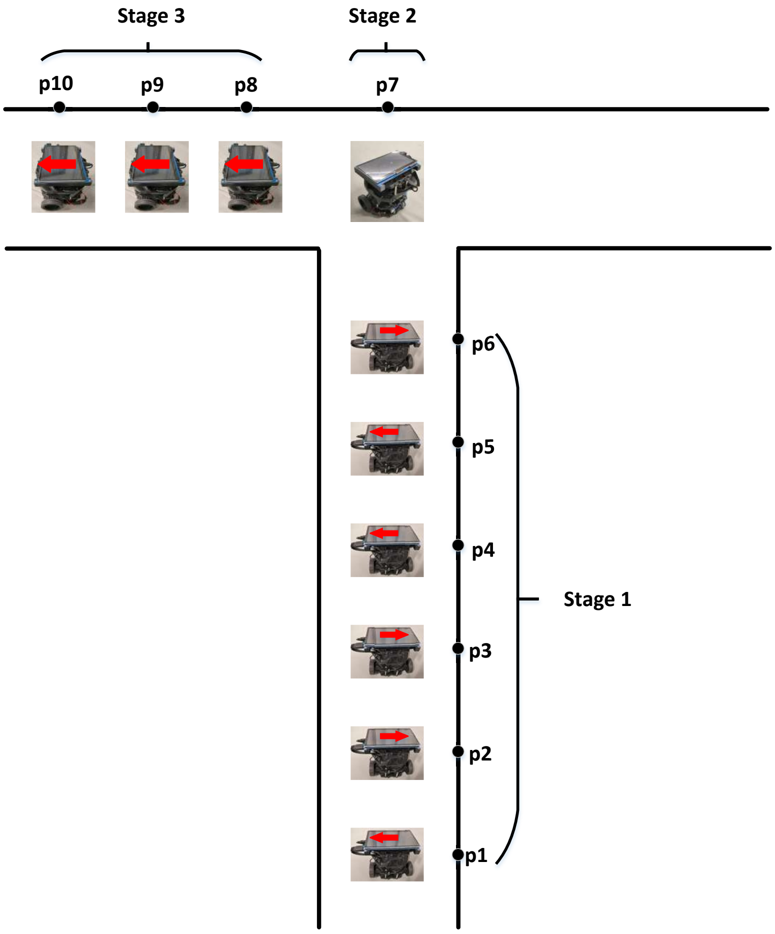


Figure 4.1: The robot signals its intentions ten times at three stages in one run. There were 6 times via LCD at the positions from p1 to p6 at Stage 1, one time at position p7 via turning movement at Stage 2, and three times at positions from p8 to p10 via LCD at Stage 3.

in the Fz channel when the robot signaled its intention via an LCD. The ERPs of correct and error conditions are similar. There is a small negativity around 150 ms, a positivity at 200 ms, a negativity at 300 ms and another broad positivity at around 400 ms, and broad negativity at around 500 ms. A statistically significant difference was found at approximately 200 ms between error and correct conditions using paired permutation tests. In (citation Testing topographic differences between event related brain potentials by using non-parametric combinations of permutation tests), permutation tests was first used to test if two ERP components had different scalp distributions.

(b) The ERP of stage 2: As shown in Figure 4.2 (b), a statistically significant difference was found at around 150 ms and 250 ms between correct and error conditions. However, compared with that at Stage 1, the peak latency of the ERP in Stage 2 was not clear and had a delay. Additionally, the ERP waves were less smooth at this stage. We hypothesize the poor observability of ERP waves was due to the ERP not precisely locking the robot turning action.

(c) The ERP of stage 3: As shown in Figure 4.2 (c), similar to Stage 1, there is a small negativity around 150 ms, a positivity at 200 ms, a negativity at 300 ms and another broad positivity at around 400 ms and broad negativity at around 500 ms. There are statistical differences at approximately 150 ms, 200 ms, and 300 ms between correct and error conditions.

We also compare the ErrP waves of stage 1 and stage 3. As shown in Figure 4.3, we observed similar ErrP waves in these two stages but the amplitude was different. Here a statistically significant difference was found at the positive peak at approximately 200 ms and the broad negative peak at approximately 500 ms between Stage 1 and Stage 3 ($p<0.05$, paired permutation tests).

Figure 4.4 shows the 64-channels scalp potentials topographies of the difference between correct and error conditions at the main peak time for one of the participants: the first is a frontal-central negativity, followed by a frontal-central positivity, and then a frontal-central negativity at the three stages.

4.2.1.2 Event-related potentials for each sequence

At each stimulus sequence, the robot constantly expressed the intended target destination for the human observer. More specifically, at stage 1, the robot signals its intended

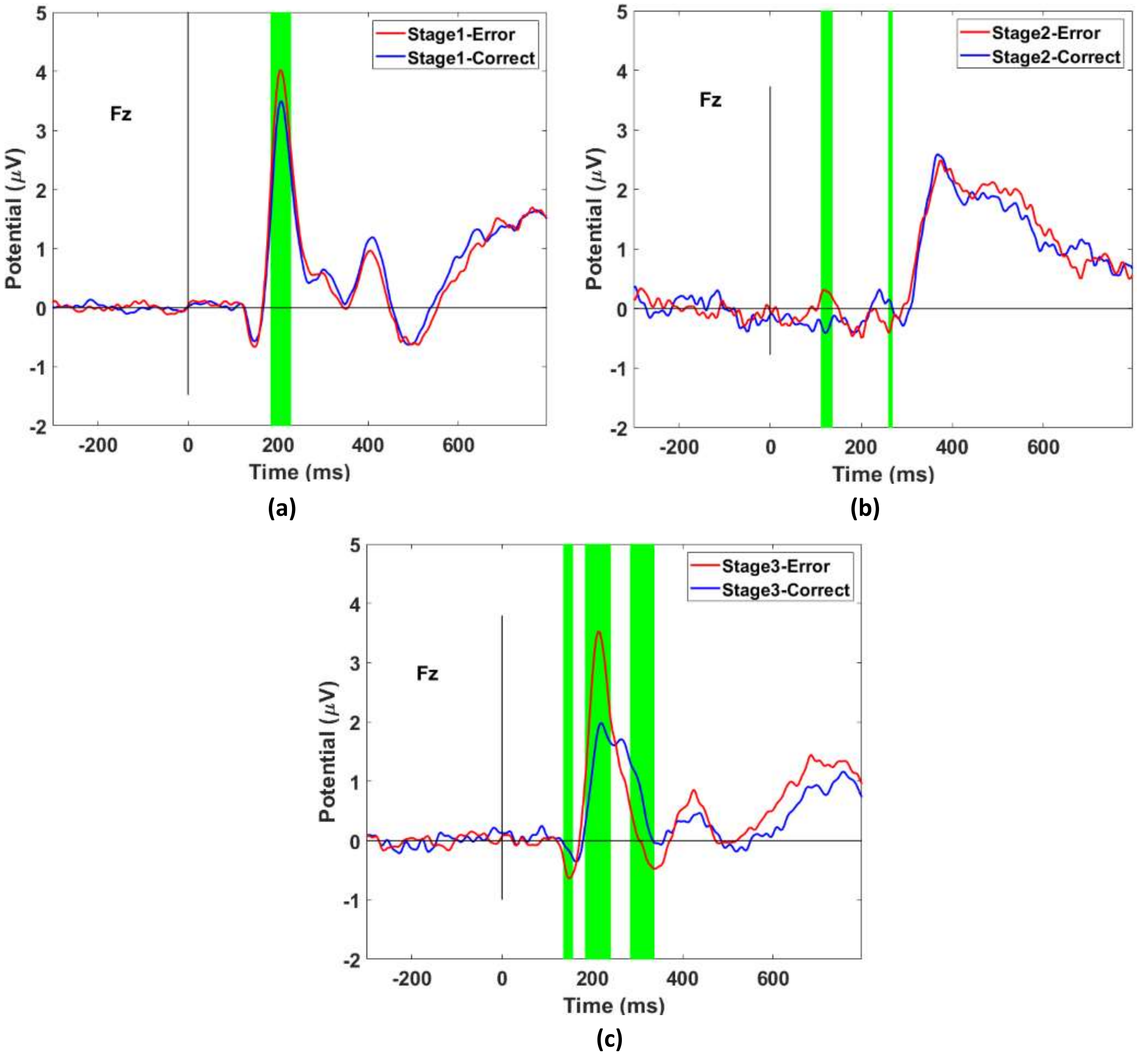


Figure 4.2: ERPs of stage1 (a), stage 2 (b), stage 3 (c)) in channel Fz. Statistically significant difference (p<0.05) was found at the green area between error and correct conditions using paired permutations test.

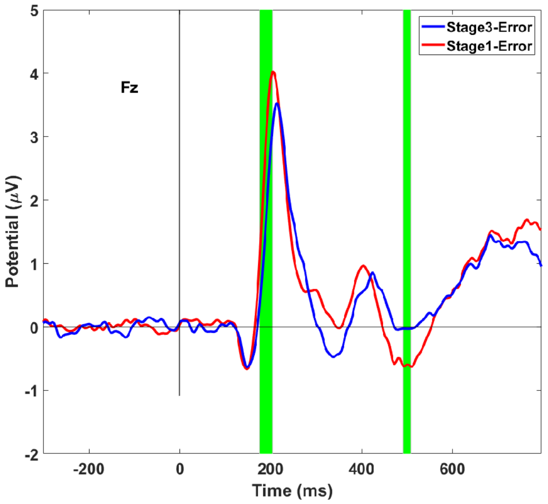


Figure 4.3: ErrP at Stage 1 and Stage 3. Statistically significant difference (p<0.05) was found at the green area between error and correct conditions using paired permutation tests.

destination with an arrow flashes on the LCD at the p1 to p6 sequence. At stage 2, the robot signals its intended target destination with either left or right turning at the p7 sequence. At stage 3, the robot signals its intended destination with an arrow flashes at p8 to p10 sequence. We explored the ERPs of each stimulus sequence of the three stages, p1 to p6 at Stage 1, p7 at Stage 2, and the p8 to p10 at Stage 3, as shown in Figure 3.2. The average ERP waveforms of the difference between the correct trials and error trials (correct trials minus error trials) of each stimulus from the Fz channel across all the participants for the three stages are shown in Figure 4.5.

(a) The ERP difference at Stage 1: A clear negative peak at approximately 200 ms could be found among the ERP waves of each sequence except for the p1 and p2 sequences. However, the positive was not clear among the sequences. The ERP waves among p3 to p6 were consistent.

(b) The ERP difference at Stage 2: No clear peak could be found of the ERP difference.

(c) The ERP difference at Stage 3: A clear negative peak at approximately 200 ms and

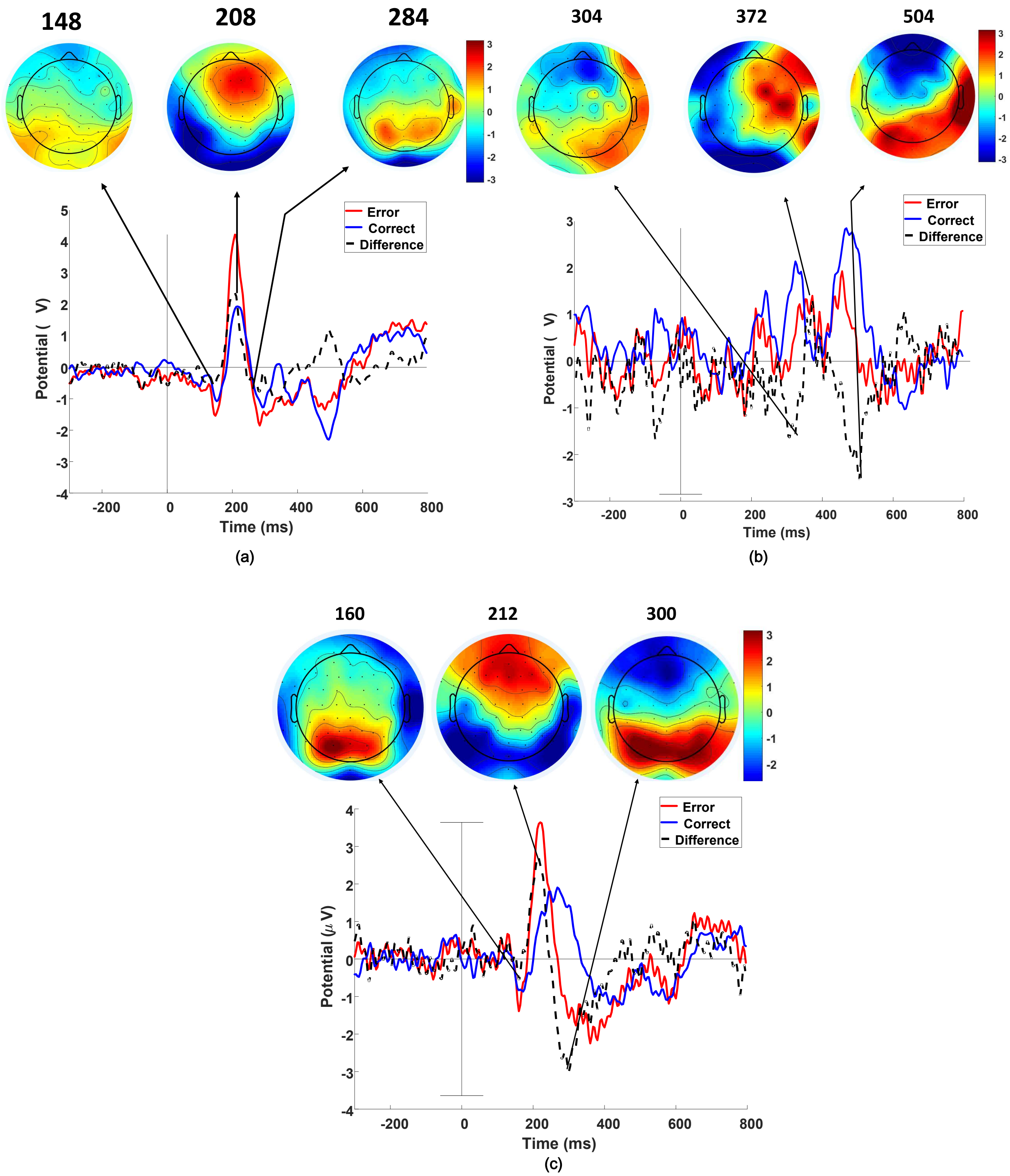
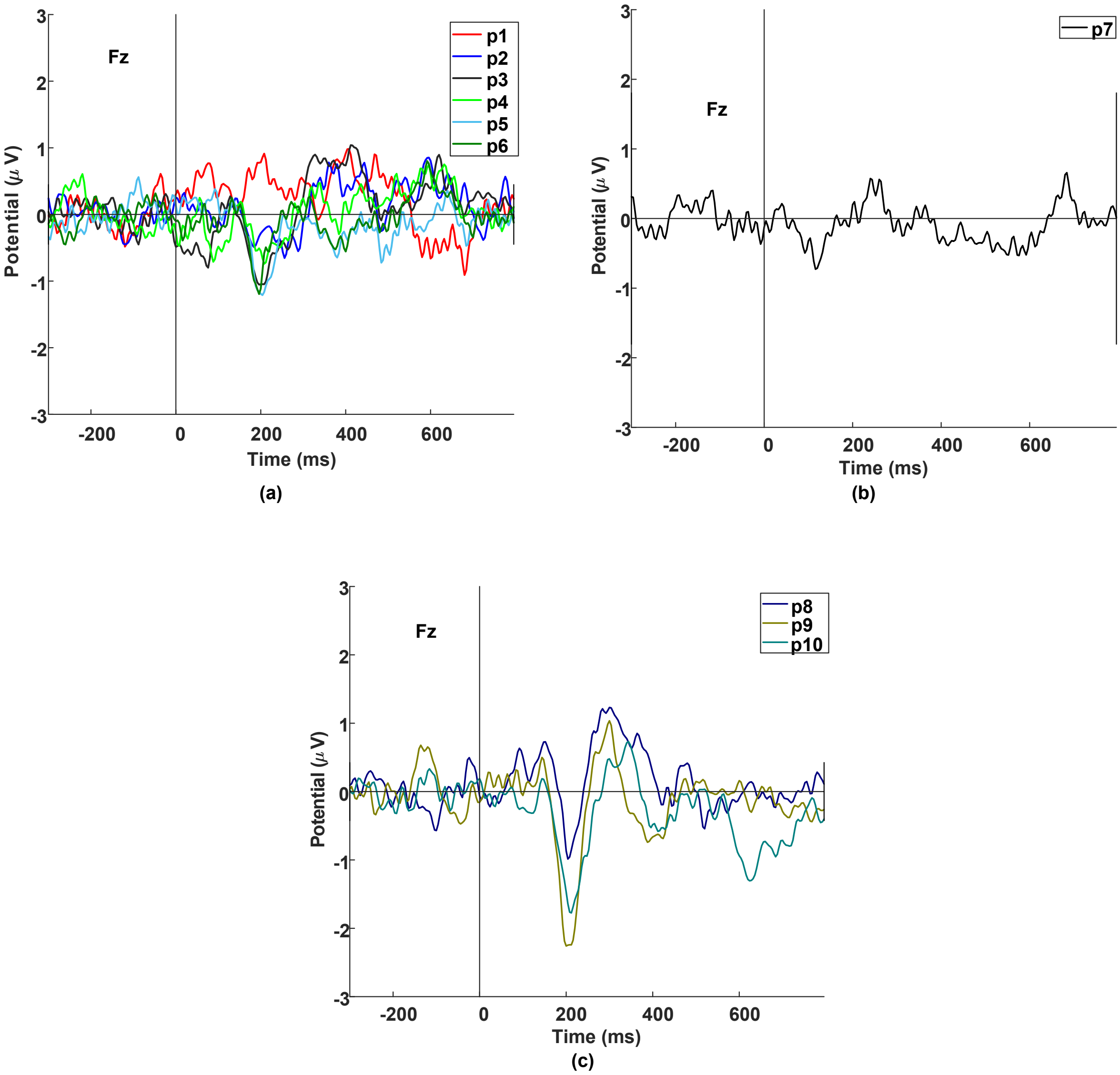


Figure 4.4: Grand averaged ERP at Fz, ERP scalp map series at certain latencies of one of the participants at Stage 1 (a), Stage 2 (b), Stage 3 (c).



(a)

(b)

(c)

Figure 4.5: Grand averaged difference in the ERP waves between the correct trials and error trials at Fz of each stimulus in Stage 1, Stage 2 and Stage 3.

a positive peak at approximately 300 ms could be found among the ERP waves of each sequence. The ERP waves among p8 to p10 were consistent in Stage 3.

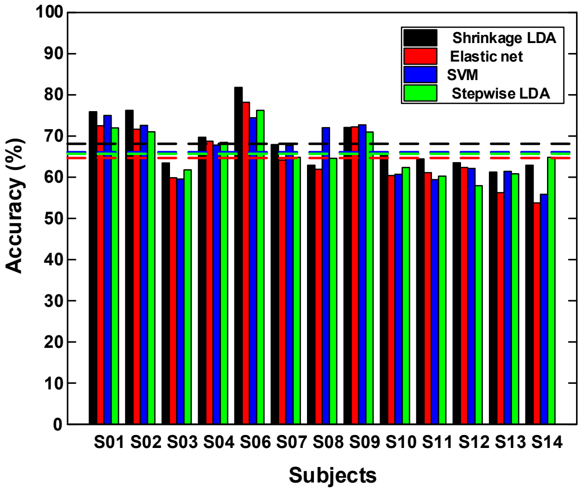


Figure 4.6: Classification accuracy of four classification methods for offline sessions.

4.2.2 Classification Analysis

4.2.2.1 Offline classification performance

(a) Offline classification result for the averaged sequence:

The accuracy (ACC) of different methods at Stage 1 can be seen in Figure 4.6. The ACC of the shrinkage LDA, elastic net, SVM, and stepwise LDA were 68.31% ± 6.42%, 64.86% ± 7.21%, 66.27% ± 6.67%, and 65.86% ± 5.45%, respectively. An ANOVA with repeated measures on the performance results revealed there was no significant difference ($\chi^2(5) = 4.348, p = 0.504$) between the four classification algorithms when combining all participants. But the receiver operating characteristic (ROC) curve of shrinkage LDA is better than the other three methods, as shown in Figure 4.7. Based on this, we choose the shrinkage LDA classifier for the following discrimination analysis.

As shown in Figure 4.8, the within-participant average ACC for Stage 1, Stage 2, and the Stage 3 were 68.31% ± 6.42%, 55.20% ± 8.85% and 69.74% ± 9.32%, respectively.

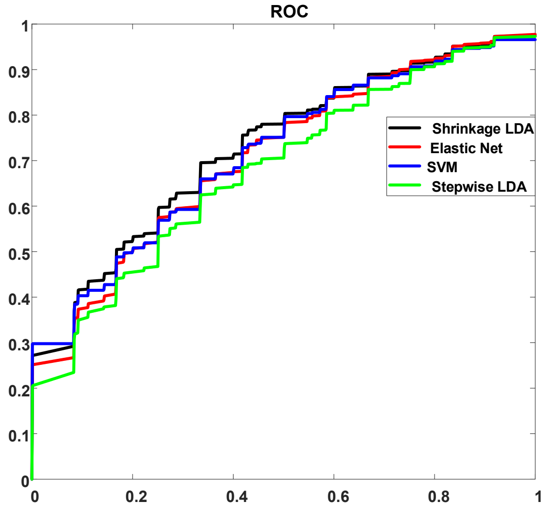


Figure 4.7: ROC curve of four classification methods for offline sessions.

(b) Offline classification result for each sequence:

In this section, we explored the effect of multiple sequences on classification performance. At Stage 1, we compared the classification performance of multiple sequences: p1, p1 to p2, p1 to p3, p1 to p4, p1 to p5, p1 to p6. Similarly, at Stage 3, we compared the classification performance of the sequence of p8, p8 to p9, and p8 to p10. As shown in Figure 4.9 (a) and 4.9 (b), the ACC increased as the sequence number increased both at stage 1 and stage 3, which coincides with the intuitive hypothesis that combining more data sequences improves the signal-to-noise ratio (SNR) for the expected ErrP signals, leading to better classification accuracy.

We also compared the classification performance of each single sequence. As in Figure 4.9 (c) and (d) shown, the ACC noticeably increases from sequence p1 to p3 and remains stable from sequence p3 to p6. In comparison, the ACCs of the three sequences of stage 3 do not change noticeably from p8 to p10. The ACC for a single sequence between participants is generally around 65% for the p3 to p6 and p8 to p10 sequences. In comparison, the ACC is around 60% for the single sequence p1 and p2. The classification performance was better when the interaction sequence

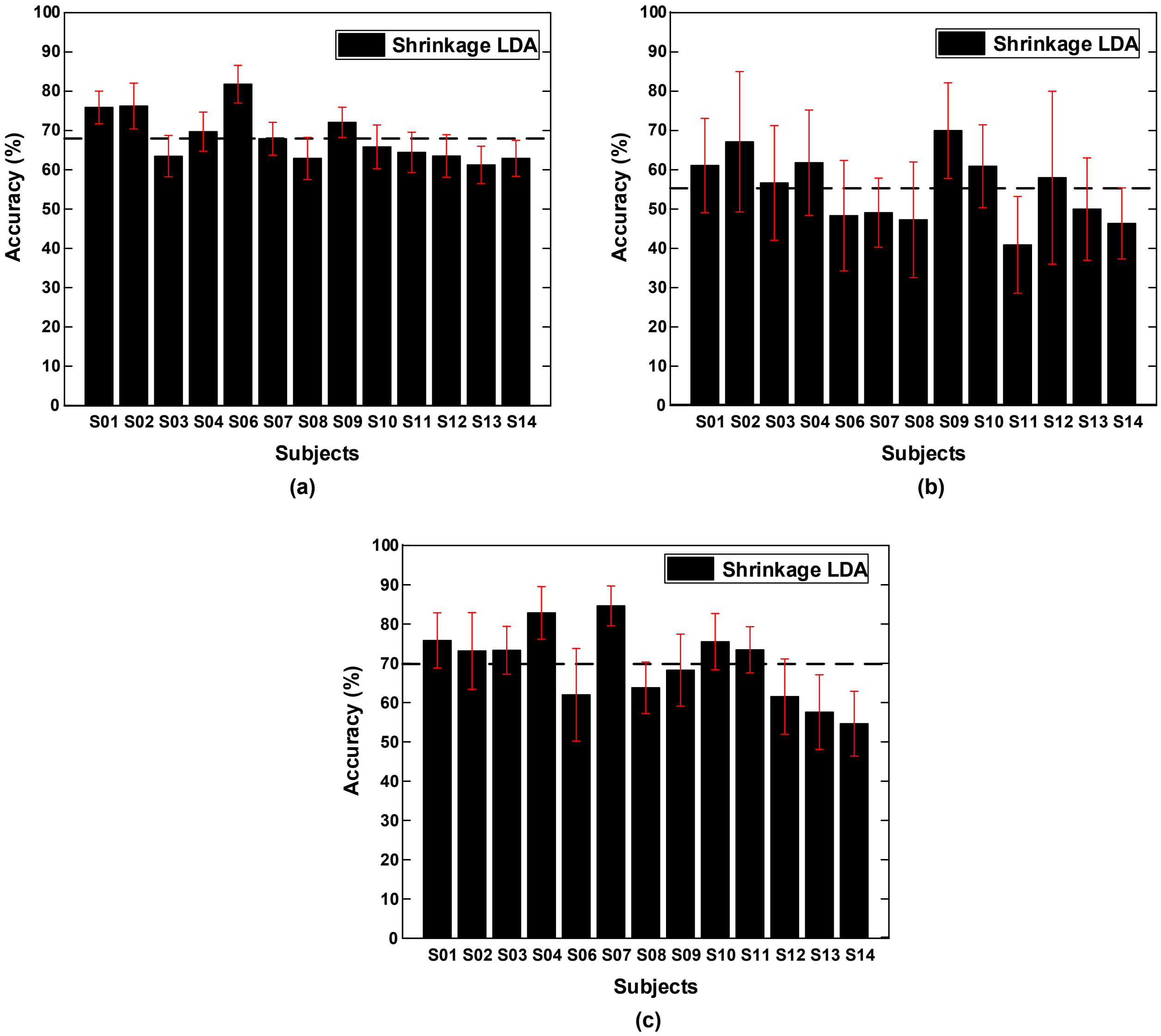


Figure 4.8: ACC for Stage 1 (a), Stage 2 (b), and Stage 3 (c).

was closed to the turning point compared with the start point at stage 1.

Considering the superior classification performance of sequence close to the turning position at Stage 1, we explored classification performance based on inverse sequences: p6, p6 to p5, p6 to p4, p6 to p3, p6 to p2, and p6 to p1. As shown in Figure 4.9 (e) and (f), the ACC first increased, reached a peak when the sequence was p6 to p3, and then diminished at Stage 1. The ACC increased as the sequence number rose at stage 3.

Instead of decoding EEG signals on a single-sequence basis, we used majority voting for multiple interaction sequences decoding for Stages 1 and 3. More specifically, at stage 1, if the number of correct decoding sequences is greater than three out of the six sequences, then the human intention could be correctly recognized. The same approach for Stage 3: if the number of correct decoding is greater than two out of the three sequences. At Stage 1, the ACC was $72.30\% \pm 9.67\%$ and $77.57\% \pm 5.70\%$ when using the majority voting procedure for the six sequences p1 to p6 and the three sequence p4 to p6, respectively. At Stage 3, the ACC was $72.89\% \pm 11.34\%$ when voting using all three sequence p7-p9. The ACC can be significantly improved by voting for the multiple sequences in one run compared with single-sequence classification. We discovered that majority voting for multiple interaction sequences can significantly improve the classification performance when using the last three interactions p4 to p6 in Stage 1.

4.2.2.2 Online classification performance

This session presents the real-time robot control strategy and online closed-loop classification performance. EEG signals from the human observer are decoded in real-time to feed to the classifier model for the robot control. The classifier model was trained using EEG data from the first three offline sessions.

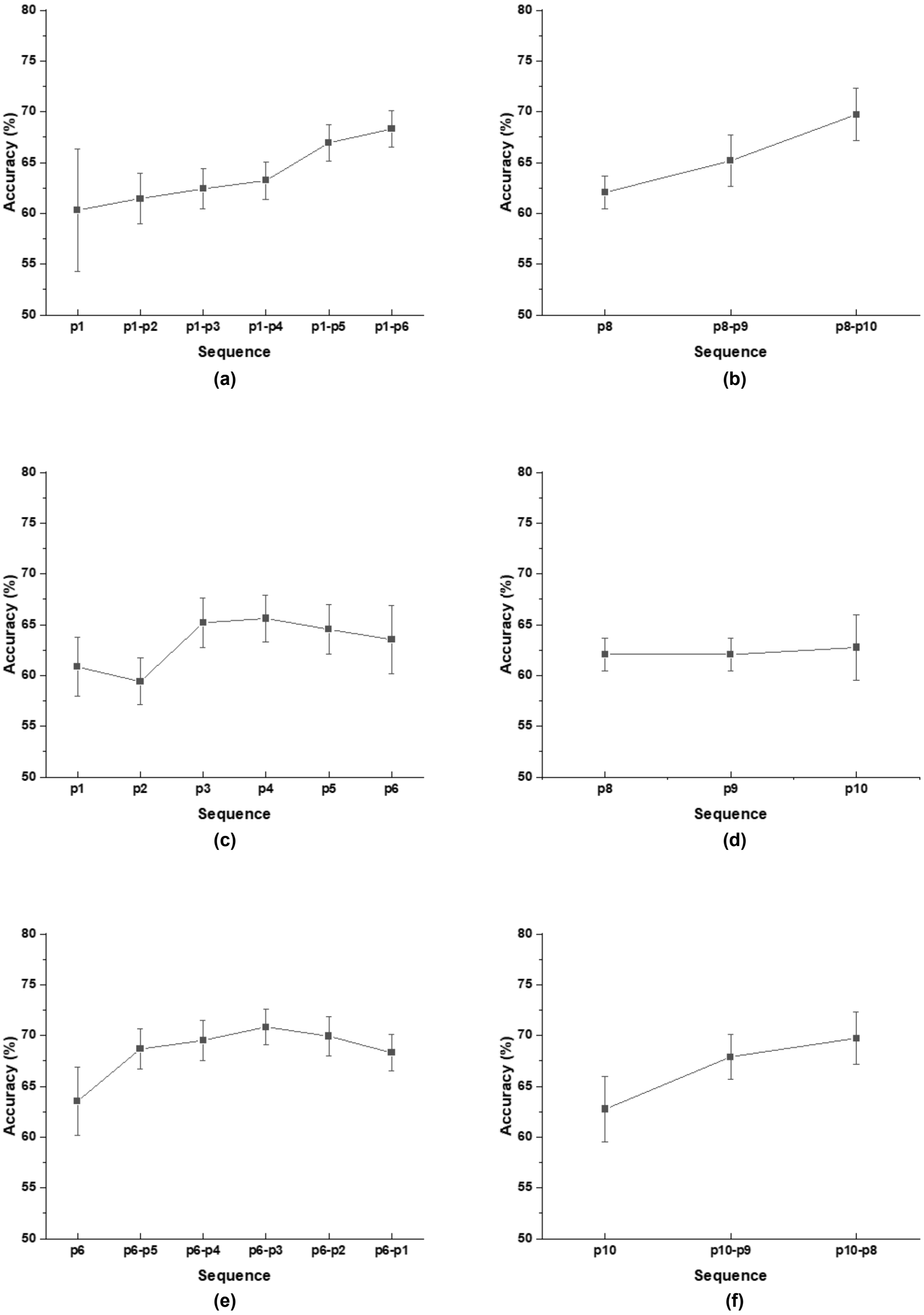


Figure 4.9: ACC of the multiple sequence of Stage 1 (a) and Stage 3 (b). ACC of the single sequence of Stage 1 (c) and Stage 3 (d). ACC of the inverse multiple sequence of Stage 1 (e) and Stage 3 (f).

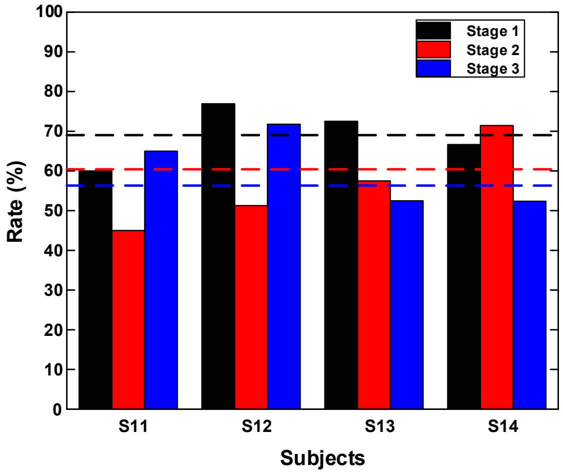


Figure 4.10: The robot correct rates at three stages of the online session.

Considering that better performance was achieved for the interaction sequence composed of p4, p5, and p6 instead of the interaction sequence consisting of the first three sequences , the interaction sequences p4 to p6 at Stage 1 combining the majority voting strategy were decoded to correct the turning direction command at the intersection. The interaction sequence p7 at Stage 2 was decoded to correct the turning direction at the interaction once a misclassification of EEG signals at Stage 1 occurred. The interaction sequences p8, p9, and p10 at Stage 3 combining the majority voting strategy were used to correct the moving direction once a misclassification of EEG signals at Stage 2 occurred.

As shown in Figure 4.10 , the correct rates to successfully decode human intention at three stages were 69.02% ± 7.33%, 56.30% ± 11.30%, and 60.42% ± 9.62% respectively. Figure 4.11 further breaks down the performance of Stage 2 and Stage 3 into true-positive rates (TPRs) and false-positive rates (FPRs). In this context, true positive can be integrated if the robot intention deviates from human expectation, then ErrP is detected from human observer. False positive can be integrated where in the human agreed with the robot intention, and no ErrP was detected from the human observer. The TPR of Stage 2 and Stage 3 was 81.74% ± 18.57% and 75.60% ± 8.46%, respectively, while the FPR was 45.73% ± 8.89% and 51.61% ± 16.57%, respectively, averaged for all participants.

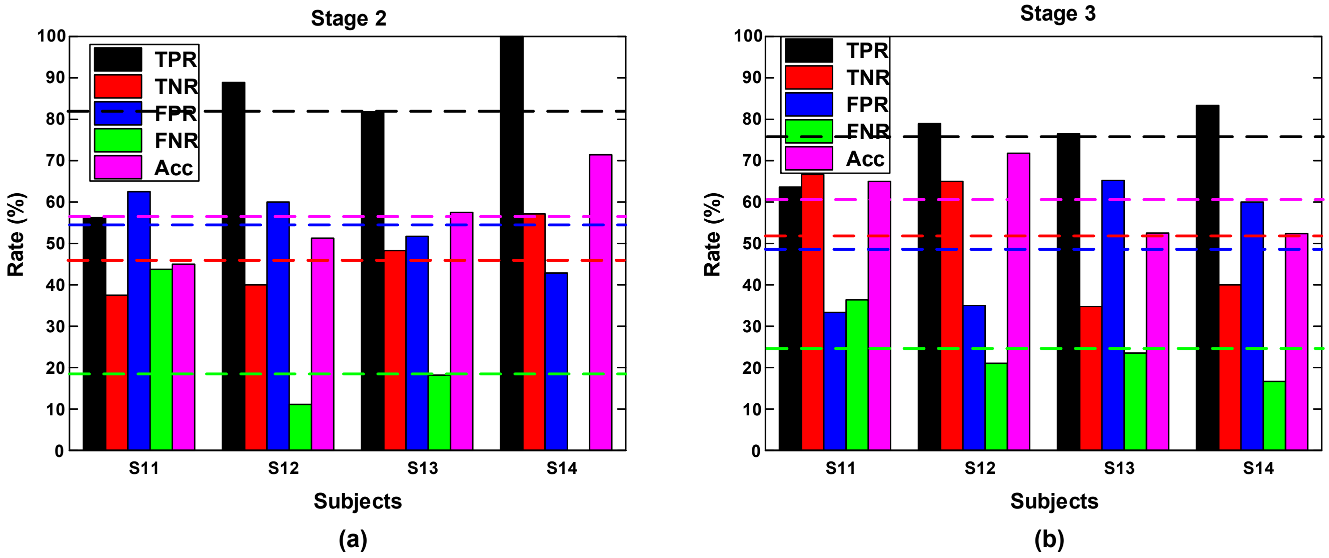


Figure 4.11: True positive and true negative rates at Stage 2 (a) and Stage 3 (b) of online session.

4.3 Discussion

4.3.1 ErrP observability and decodability for evaluating of the robot's intention

Most of the proposed error-based HRI systems have used ErrP as implicit input to correct machine errors after an error happens. The error might be costly or dangerous in a context such as an assembly line or autonomous driving. However, in our proposed system, the robot signaled its intention to the human before making its decision. ErrP signals would be elicited and fed to the decoding model in real-time to generate control commands if the observer disagreed with the robot's intention. Thus, the observer can evaluate the robot's intention at any time via an additional visual channel and intervene early, if necessary, before the robot commits an error. At Stage 1, the LCD attached on the ground robot constantly communicated the intended target destination with left or right arrow flashes. The EEG signals were decoded to generate a turning command at the intersection. During Stage 2 the robot expressed the intended target destination with either left or right turning at the intersection. The EEG signals were decoded to correct the turning direction at the interaction point once a misclassification of EEG signals of stage 1 occurs. At Stage 3, the LCD communicated the intended target destination with arrow flashes with the direction either directing or deviating the target. The EEG signals were decoded to correct the moving direction once a misclassification of EEG signals of stage 2 occurred. The high ACC of the single-trial ErrP experiment demonstrated the feasibility of ErrP evoked by the intention discrepancies between the human and the robot in the HRI system, which widened the usage of ErrP in traditional HRIs.

The averaged ERP observability across all participants, and single-trial classification performance was consistent between Stage 1 and Stage 3, as shown in Figure 4.2 and Figure 4.8. There is a small negativity around 150 ms, a positivity at 200 ms, a negativity at 300 ms and another broad positivity at around 400 ms, and broad negativity at around 500 ms after the BCI feedback at both Stage 1 and Stage 3, which coincided with ERP waves in this study [77]. The same ERP phenomenon at Stage 1 and Stage 3 indicates that the observed brain activity at both these stages originated from the same underlying neural process of error monitoring [72] [168]. As shown in Figure 4.3, we observed similar ErrP waves at Stage 1 and Stage 3, but the peak amplitude was statistically different at the positive peak at approximately 200 ms and the broad negative peak at approximately 500 ms. We hypothesize the peak amplitude difference was due to the stimuli distance from the human observer. This is because event-related potentials are strongly related stimuli size [189]. In particular, the stimuli at Stage 1 was closer to the human observer than those at Stage 3, which could elicit strong ERP waves.

In most studies, ErrPs were observed after an error event occurred. The terms 'primary ErrP' and 'secondary ErrP' were first reported in [148]. The authors discovered that a secondary ErrP was evoked once a misclassification of the EEG signals occurred. However, the secondary ErrP was not used in real-time due to an ERP time-locked issue.

In our paper, for the online close-loop session, the robot intention at stage 2 and stage 3 can reflect the classification results of stage 1 and stage 2, respectively. Real-time decoding of the observer's EEG signals at stage 2 and stage 3 allows continuous correction of misclassification.

Previous research [47] explored the effect of 20% and 50% error rates on the ErrP. They discovered that the ErrP amplitude was small when the error rate was 50% compared with that when the error rate was 20%. In our proposed method, the correct and error rate feedback of robot intention was 50% rate. The user needs to mentally judge the robot's feedback was correct or not without any other physical response, which can increase the difficulty of the ErrP discrimination. Additionally, it has been shown that the classification performance of continuous BCIs was worse than that of noncontinuous BCIs [161]. It was hypothesized that the continuous interaction would likely lead to a higher workload than standard BCI systems. The ERP amplitude diminishes with a high brain activity workload, which could affect the classification performance during continuous interaction. Continuous interactions could also affect the decodability of ErrP in our system. Another impact was the moving stimulus, which could affect single-trial variability, affecting the ErrP decodability.

As shown in Figure 4.11, the TPR of Stage 2 and Stage 3 was 81.74% ± 18.57% and 75.60% ± 8.46%, respectively, while the FPR was 45.73% ± 8.89% and 51.61% ± 16.57% averaged all participants, also respectively. The high TPR indicates that ErrP could be actively evoked if the robot intention deviated from the observer's expectation. In contrast, the low FPR indicates that the human observer tends to "over" react to the robot's intention even though it was correct.

Another reason for the high TPR and low FPR may be the different levels of human observer engagement between online and offline sessions. The decoder used for online control was trained using offline data. The human observer engages more during the online session than during offline sessions, since the lack of collaboration during offline sessions significantly affects the human observer's engagement. ERP amplitude and latency were related to the participant's attentional involvement [19]. Online adaptive learning would be a useful approach to adapt the variation between offline and online sessions [17]. The new online data will be utilized to retrain the model obtained from offline sessions and hence improve it iteratively. However, the findings of the online performance were based on only four participants. Future research could be conducted in this direction with more participants.

In our paper, the ErrP decoder was a binary classifier, and the output corresponded to the robot's binary intention (left or right). Our experiment was based only one dimension (left and right) at the intersection. The robot's intention (left or right) could be decoded with the ErrP-based binary classifier. However, one important question is how ErrP could be decoded and used to control the robot's movement when the map dimension increase from one- to two- dimensional direction, for example, left/right and forward/back in a maze. Further research could be undertaken in this direction.

4.3.2 Using robot motion as time-locked events for ErrP

As shown in Figure 4.2 and Figure 4.8, both the classification performance and ERP waves of the robot action feedback at Stage 2 are poor compared to that of the arrow flashing feedback at Stage 1 and Stage 3. In addition, the classification performance variation was higher across participants when the stimulus was the robot turning movement at Stage 2 (see Figure 4.8). This finding is consistent with that reported in [37], where the ErrP was sensitive to the stimulus despite identical protocols. We believe the poor observability of the ERPs in Stage 2 is because the perceived robot turning movement onset is not precisely time-locked compared to the flashing arrow stimulus.

To study the time-locked issue, we designed an additional experiment that examined the human reaction time toward the robot's turning motions. In each trial, an arrow will randomly flash with the direction either left or right. After 2 seconds, the robot will randomly perform a rotation with direction either left or right. We instructed the user to press the key "v" when she perceived the left turning motion of the robot and to press the key "b" when she perceived the right turning motion. The experiment consists of 100 trials with 50 left turns and 50 right turns. We recruited two participants for the experiment. The average time for the user to perceive the turning motion was 591.8 ± 117.6.5 ms. The initial result suggests that the selected epoch period, 150 - 600, for the ErrP analysis and decoding is reasonable. The large variance response time toward robot turning also confirmed our assumption that the onset of turning motion could not be precisely time-locked.

The same time-locked issue was also reported in [148], where the terms 'primary ErrP' and 'secondary ErrP' were first reported. The primary ErrP was precisely locked by a mechanical momentary switch affixed to the robot. However, the secondary ErrP was not used in real-time control due to an ERP time-locked issue. It should be noted that the time-locked issue is the limitation of ErrP-based BCI. The feedback onset time is vital for reliable EEG classification, which needs to be carefully considered for researchers while designing an ErrP-based BCI experiment.

4.3.3 Interaction sequence effect on ErrP

Given that existing BCI paradigms cannot achieve high precision using one single visual stimuli, the proposed experiment was designed to explore the optimal number of visual stimuli and time period for the ErrP-based implicit robot control. The ERP activity and classification performance of the p1 and p2 sequences were poor compared with those of the p3, p4, p5, and p6 sequences. A possible reason is that the user was more involved in the task when the robot was close to the turning point. The poor classification performance of the p1 and p2 sequences are made evident by the single sequence ERP waves, as shown in Figure 4.5. No obvious negative peak at approximately 200 ms could be found in the p1 and p2 sequences. This was consistent with the low classification rates for the p1 and p2 sequences. In contrast, a clear negative peak could be found for the p3 to p6 and p8 to p10 sequences. High classification rates seemed to reflect large N2 components. The ACC for every single sequence across participants is generally around 65% if the corresponding N2 negativity could be clearly seen.

As shown in Figure 4.9 (a) and (b), the ACC increased as the sequence number increased in both Stage 1 and Stage 3. One potential reason is that the multiple sequences improved the SNR for the ErrP signals. Another potential reason is related to the single-sequence difference, especially in Stage 1, as illustrated in Figure 4.9 (c). The classification rate increased, reached a peak with the sequence p6 to p3, and then decreased as the inverse sequence due to the poor performance of p1 and p2 sequences.

We found that the majority voting across the classification result for multiple sequences can improve classification performance. In particular, when using the voting procedure for the classification result of the last three sequences in Stage 1, the ACC reached 77.57% ± 5.70%. It demonstrated that the voting procedure is an effective way to decode implicit human intention with multiple interaction sequences. This finding was consistent with what was reported in [100], in which multiple events analysis can increase the accuracy of error detection.

Chapter 5

Evaluation the feasibility of shared autonomy with simulated ErrP

This chapter presents the experiment overview and results analysis from the experiment design in Chapter 3, Section 3.2. Later parts of this chapter will discuss what those results mean.

5.1 Experiment overview

The main purpose of this experiment is to explore the feasibility of shared autonomy, which combines user input and agent observation to build a shared autonomy control system. As shown in Figure 3.3, we combine user input with simulated ErrP and robot observation as inputs of the deep model to learn a shared control strategy. However, due to the low decodability of ErrP, A direct mapping of partial observations to action is not sufficient for optimal behavior. To describe the uncertainty about ErrP, we formulate the shared autonomy as POMDP. We use the actor-critic method that relies on RNN to deal with POMDP. RNN is an approach to stacking the memory history and robust to partial observation [68].

We begin our experiment with simulated user feedback to train and test the shared autonomy model in target search task. The goal location and agent start position are randomized at each episode in a constant static map through training and testing. We hypothesis that the ErrP-based shared autonomy would enables an autonomous robot to complete the tasks more efficiently.

5.2 Results analysis

We begin our experiments by executing our method under different simulated ErrP accuracy levels in various environments. We first make ErrP as a full observation with 100% accuracy and then make ErrP as a partial observation with different accuracy levels. We prime the solo agent without ErrP feedback to function as the baseline. Our central hypothesis is that our method can improve the agent's performance despite partial ErrP observation. Simulating pilots enables us to take a deeper dive into different aspects of our method (like the effects of ErrP accuracy level on training an effective shared control model, gradient analysis with different ErrP accuracy at different positions). Most importantly, we use simulated ErrP to train the shared control model that test on real human users.

5.2.0.1 100% accuracy ErrP and without ErrP

We make the solo agent without ErrP feedback as the baseline. The agent trained without ErrP is a solo autonomous agent. We first make ErrP as full observability with 100% correct probability and compared the training and test performance with that of baseline condition to check the feasibility of the shared control paradigm. We evaluate the method on the grid environment without obstacles, as shown in Figure 3.4 (a).

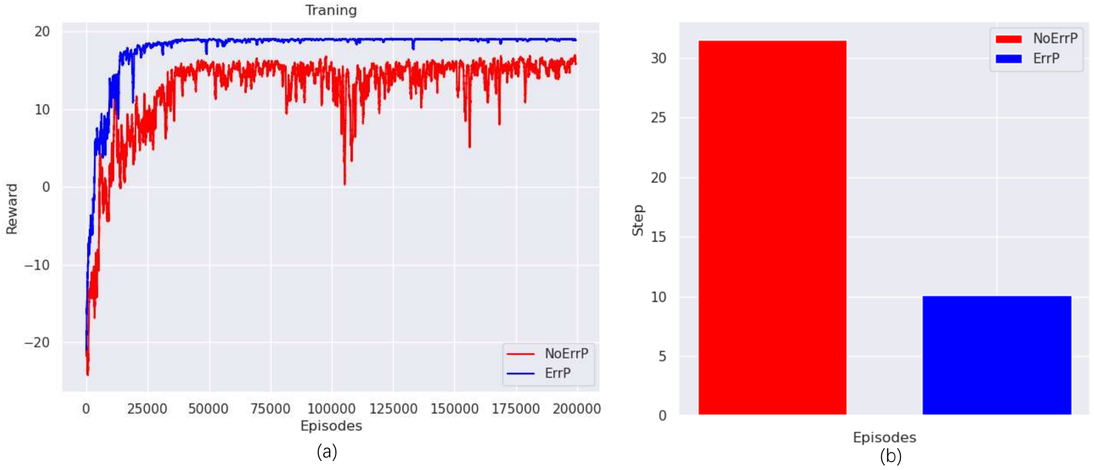


Figure 5.1: Training curve with 100% correct probability ErrP and no ErrP conditions (a). The averaged step is used to reach the target positions (b).

As shown in Figure 5.1 (a), the training convergence is faster and gains more rewards with ErrP as input than without ErrP. It demonstrated that the shared control policy does better than the solo agent policy. The learning curve is smoother with the ErrP condition, which demonstrates that the performance is more stable. There was a significant improvement in the training convergence when ErrP was used as input. The ErrP feedback is deemed to be agent's additional observation. With more observation and information gained, the agent can make better decisions. Figure 5.1 (b) was the averaged steps used during evaluation with 20000 episodes with random start and target positions. The average steps taken with ErrP and without ErrP feedback were 10 and 32 steps, respectively.

5.2.0.2 ErrP effect on maze size.

In this section, we explored the ErrP effect on different maze sizes. We hypothesize that the agent's performance would improve greatly using a large maze. We train and test the model on three different grid sizes: 9×9, 10×10, 11×11. We compared the average steps used during the test phase between 100% accuracy ErrP and no ErrP

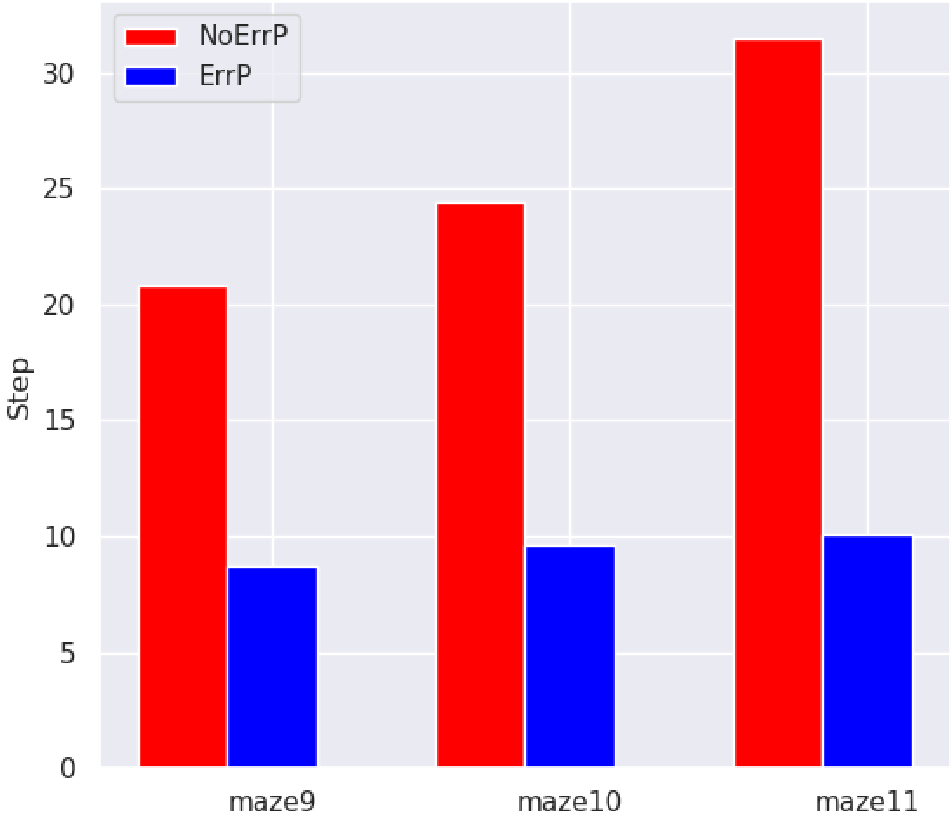


Figure 5.2: Average steps during test on different maze size for ErrP and no ErrP conditions.

conditions. As shown in Figure 5.2, the average steps were 20.8, 24.6, 31.5, respectively, for no ErrP condition and 8.7, 9.6, 10.1, respectively, for ErrP condition on the three different mazes as size increased. With ErrP as input, the agent performance concerning the average steps increased by 58.1%, 60.6%, and 68.0%, respectively, compared with no ErrP condition for the three different mazes with increased size. The results confirmed that ErrP wielded more effect on the agent's performance on the large maze than the small one. This can be interpreted as meaning that the ErrP acts as a critic of the correct or error searching direction. With the ErrP guidance, the agent can quickly find the target position.

5.2.0.3 ErrP as partial observability.

In this section, we make ErrP as partial observation by adding different noise levels. We vary the correct accuracy of the ErrP feedback with simulated data and evaluate with the same level of correct probability. The probabilities of providing the correct ErrP feedback were initialized from 0.65 to 1. We then used the simulated data to train different models: no ErrP model, a POMDP model (correct probability from 0.65 to 0.9), and an MPD model. For testing, we used 20000 episodes with random start and target position. We evaluated the different models with the same correct probability feedback. The average steps of 20000 were recorded.

Figure 5.3 (a) shows the training performance with different ErrP correct probability as well no ErrP feedback conditions. Figure 5.3 (b) depicts the averaged steps used during evaluation with different conditions. The higher the correct probability, the small steps used to reach the target position. In other words, the less uncertain feedback from humans, the better decision they can make. The agent can learn different optimal policies with diverse sets of partially observable environments. The 100 % correct probability can be considered a full MDP, while the rest with ErrP feedback can be considered POMDP. The experiment indicates that the agent can infer the correct probability during training and learns an optimal policy. This indicated that integration of agent observation and human perception could greatly improve the search efficiency; even the human feedback is partially inaccurate during the continuous search task. Furthermore, we found that when the correct probability was lower than 65 %, more steps were used to reach the target position than that without the ErrP condition. The less correct probability was not good enough to train an optimal shared policy compared with that solo agent policy. In other words, the uncertain human feedback will mislead the agent's decision.

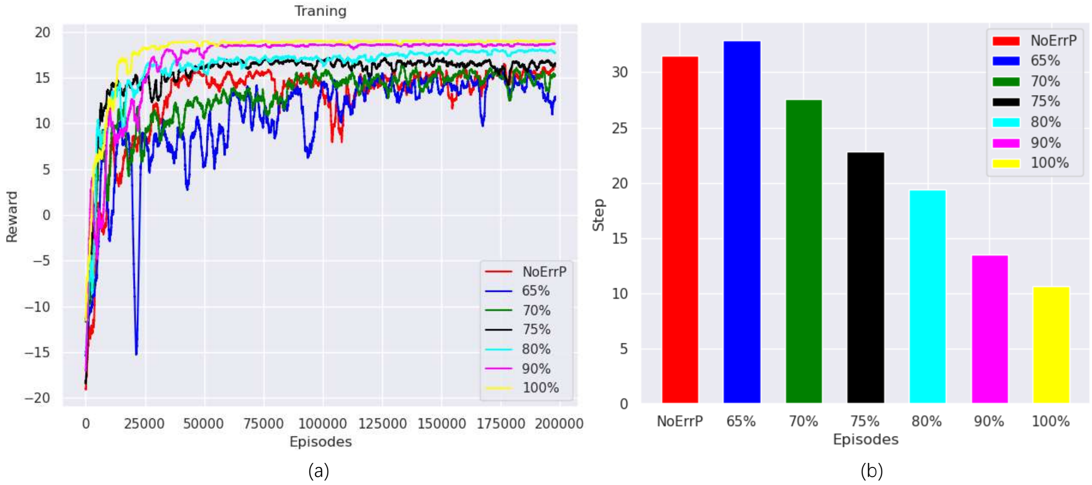


Figure 5.3: Training curve with different level probability ErrP and no ErrP conditions (a). The averaged step is used to reach the target positions (b).

5.2.0.4 Search policy explanation

We examine the search trajectories while fixing the agent's start location to understand the model strategy. In the shared control policy with 100% correct probability, we found that if ErrP appears, the agent will change its search direction to its left so that takes up anticlockwise searching, as shown in Figure 5.4. The same performance when ErrP accuracy was bigger than 80%. The agent will always change its search direction on the left-hand side. In other words, the policy trained with 80% ErrP accuracy was the same as that of 100% ErrP accuracy. However, the agent learned a different accuracy when ErrP accuracy was less than 70%. The agent would not change its search direction immediately if ErrP occurs. The agent will change its direction when it gains more confidence. We hypothesize that the confidence has relation to ErrPs of past several steps and current position. Thus, the confidence the function:

$$C = f(E(n),(x,y))$$

Here we use E(n) as the effect of past n steps ErrP.

We also checked the search pattern that trained on the environment with obstacles, as shown in Figure 3.4 (b). For the environment without obstacles, the agent will change its search direction to the left-hand side once there is an ErrP input, other than returning to the last position, which learns an inverse spiral search strategy. However, for the environment with obstacles, the agent will first return back to its last location and then change its direction to the left if there is an ErrP input. The difference was that returning to the last position for the environment with obstacles was more steady for the agent. The agent has the potential to be blocked by the obstacles in the following steps if it is searched in an inverse spiral way.

For the no ErrP model, as shown in Figure 5.5 the agent's trajectory takes on of anticlockwise searching. The agent searches along the wall and then narrows the search region. The trajectory is fixed and depending only on the agent's start point. While for the ErrP condition, the agent changes the trajectory in real-time to adapt to human feedback.

5.2.0.5 Trained with full observation and evaluated with partial observation

In order to test the robustness to the uncertainty of the POMDP model, we compared two conditions: 1) The agent was trained with partial observation (75% ErrP correct probability) and evaluated with incrementally more complete observation; and 2) The agent was trained with full observation (100% correct probability) and evaluated with an incrementally more complete observation from 70% correct probability to 100% accuracy. Figure 5.6 show the averaged steps used during the evaluation of different situations. The averaged steps were reduced when the correct probability increased for both two conditions. However, when the correct probability is less than 80%, the model trained with partial observation used fewer steps than with full observation. Conversely, when the correct probability is greater than 80%, the model trained with partial observation used more steps than full observation. The POMDP model allows performance to scale

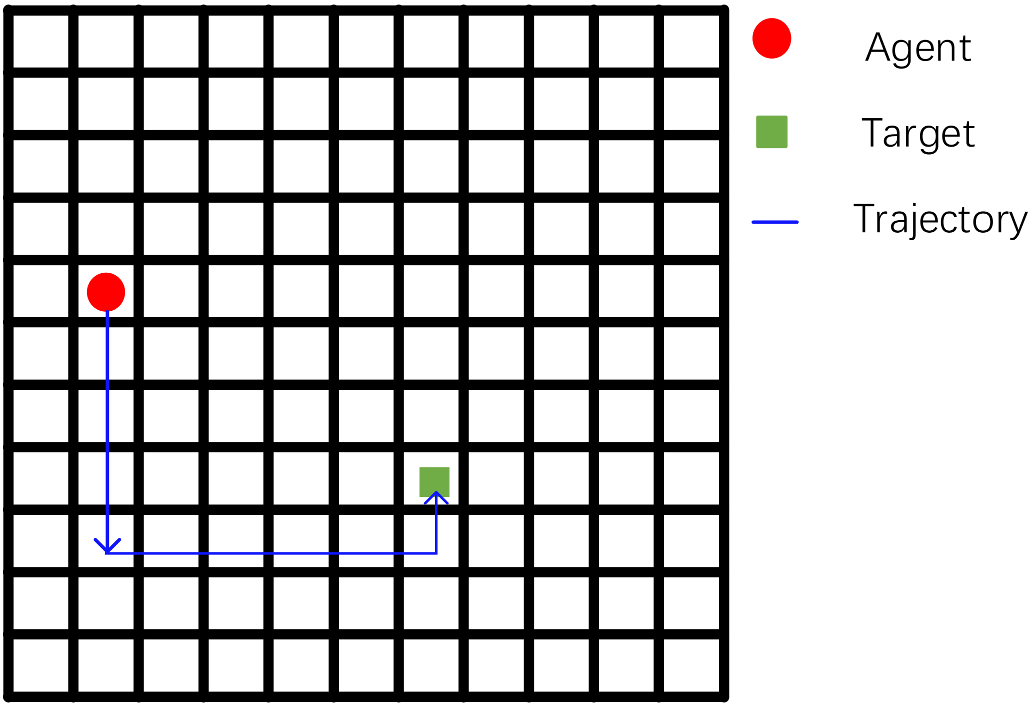


Figure 5.4: The agent search policy with 100% accuracy ErrP.

linearly as a function of observation quality. Note that, when evaluated with 70% correct probability, while both models incur a decrease in performance, the MDP model dropped to around 40 steps and the POMDP model dropped to approximately 27 steps. Meanwhile, the performance of the model trained with full observation declined dramatically when given incomplete observations. When evaluating with 100% accuracy, the POMDP model used around 12 steps, reaching the near-perfect levels (around ten steps). In all, the model trained with partial observation is more able to handle partial observability when the quality of observations changes during evaluation time. The performance declines dramatically when using the model trained with full observation and tested with partial observation. The results are consistent in [68]: the model trained with partial observation is robust enough to handle missing game screens and scalable enough to

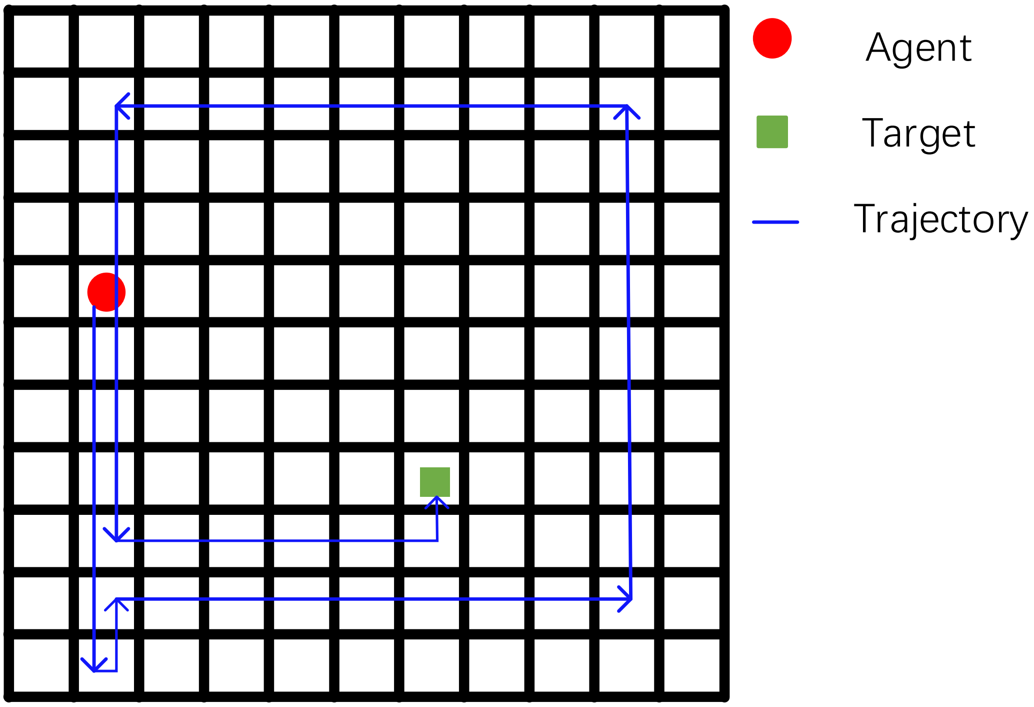


Figure 5.5: The agent search policy without ErrP.

improve performance as more data becomes available.

It demonstrated that the model trained with partial observation was more robust to uncertainty during evaluation, despite the learned two models using the same neural network architecture. Besides, it is also scalable enough to improve performance as observation of correct probability increases.

5.2.0.6 Gradient analysis on ErrP with different observation level

The gradient is computing derivatives of the outputs of a model with respect to the input's variables, which identifies which input variables are important for the prediction of the outputs. The gradient-based method is a natural and popular attribution way [198] to explain deep neural network decisions. It indicates how important the input dimension is for the output based on the learned model.

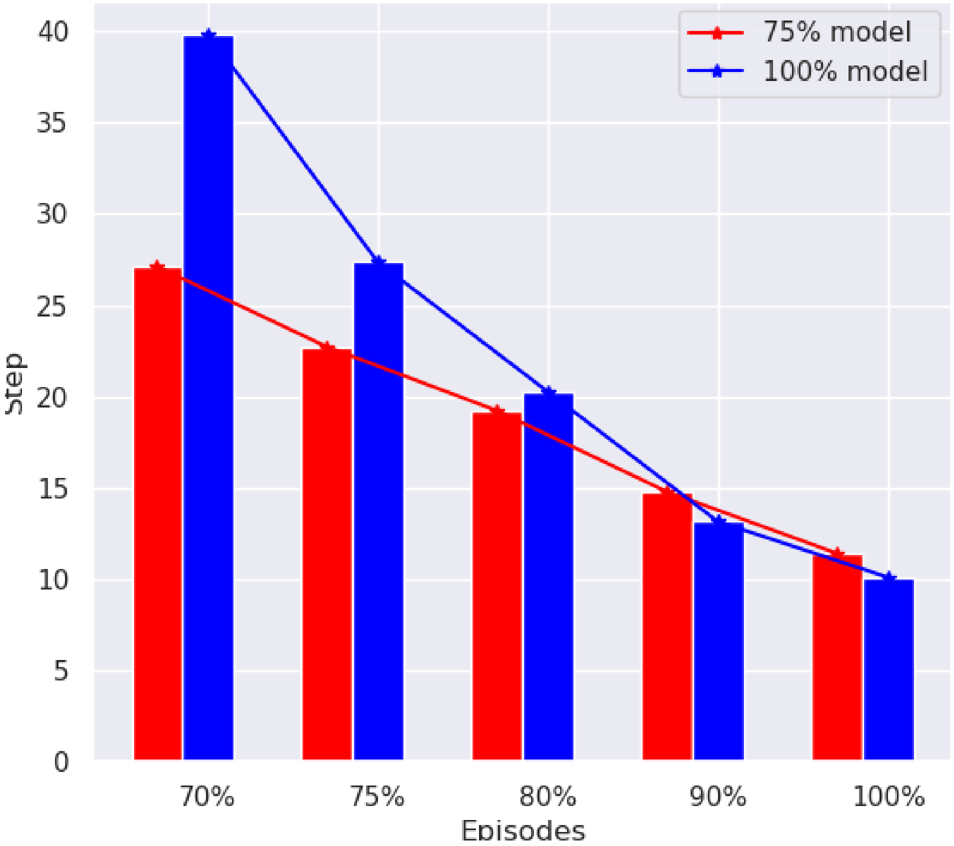


Figure 5.6: The averaged steps to reach the target position on different observation levels with partial and full observation trained model.

To better understand the mechanisms that allow the POMDP model to perform well in the uncertain environment, we analyzed the performance and gradient between a POMDP model at different correct probabilities. More specially, we compared the gradient with 70%, 75%, and 80% correct probability. We found that when the ErrP correct probability is bigger than 80%. The learned policy is the same as that of 100% probability. That means the ErrP gradient makes no difference when the correct probability is bigger than 80%. So, we simply compared the gradient trained with a correct probability lower than 80%.

As shown in Figure 5.7, ErrP gradient increased as the ErrP correct probability increased. It revealed that when the ErrP accuracy is higher, it has more important effects on the outputs than low ErrP accuracy. In contrast, with the large effect on the

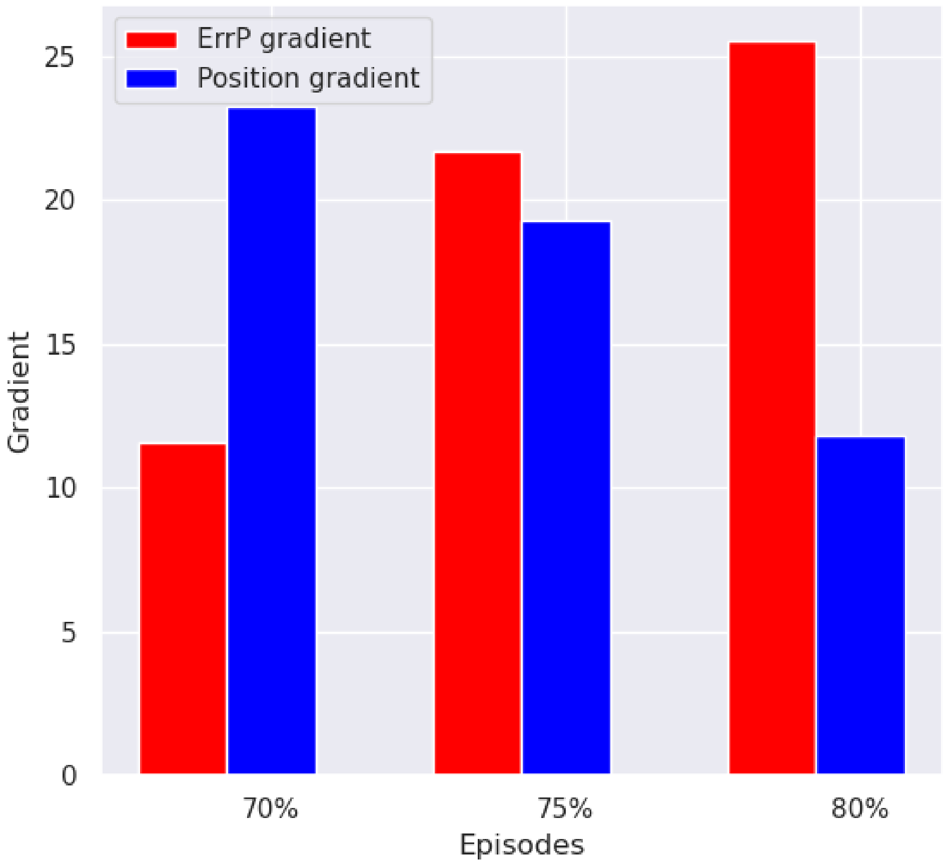


Figure 5.7: ErrP and gradients with different ErrP correct probability.

output of ErrP, the effect on the output of other inputs variables should abate. In other words, the gradients of the agent's observation, like position variables, will decrease. The position gradient decreased as ErrP accuracy increased. The results demonstrated that human feedback gradually has more effects, and the agent observation has fewer effects with the ErrP accuracy scale increased during training.

It revealed that the model trained with high ErrP accuracy was more sensitive to ErrP input. Corresponding to the behavior, the agent will more likely change the search direction whether there is negative feedback from a human with the model trained with high ErrP accuracy. In other words, human users have a more significant effect on the agent's action with the model trained with a higher ErrP accuracy than less accuracy.

5.2.0.7 ErrP gradient analysis on different positions

During the test phase, we found that ErrP has more effect on the central area outputs than in the edge area. We visualized the ErrP gradient map of the model of the two environments to check whether ErrP has different effects on outputs at different positions. The computation of the gradient map is extremely quick since it only requires a backpropagation pass. The gradient map encoded the importance of the ErrP effect on the agent's action at different locations. The use of color represents the gradient value.

Figure 5.8 is the model gradient map of two maze environments. The color indicates the gradient value at different positions. Generally, the gradient is big in the central area, which means that the ErrP significantly affects the critical position. Meanwhile, the gradient is lower at the edge areas, which means the ErrP feedback has no noticeable effect on the agent while searching along the wall.

The initial ErrP gradient analysis showed that ErrP has more effects on the critical location. In future research, the more advanced interpretation methods such as Integrated Gradients [163], SmoothGrad [160] could be used for more detailed analysis.

5.3 Discussion

5.3.1 Formula the learning as POMDP with noise ErrP

The policy learned with clean observation (100% accuracy ErrP) is not feasible (vulnerable) when the environment is inherently noisy during the test. The discrepancy between clean simulated ErrP data and real human EEG data contributes to the "reality gap." The real human actual ErrP feedback can not 100% match the simulated one. The shared policy may fail in the real environment due to the low decodability of the ErrP signal. We formulate the learning as POMDP and train the model with simulated noise observation during training. We discovered that the model trained with partial observation is more

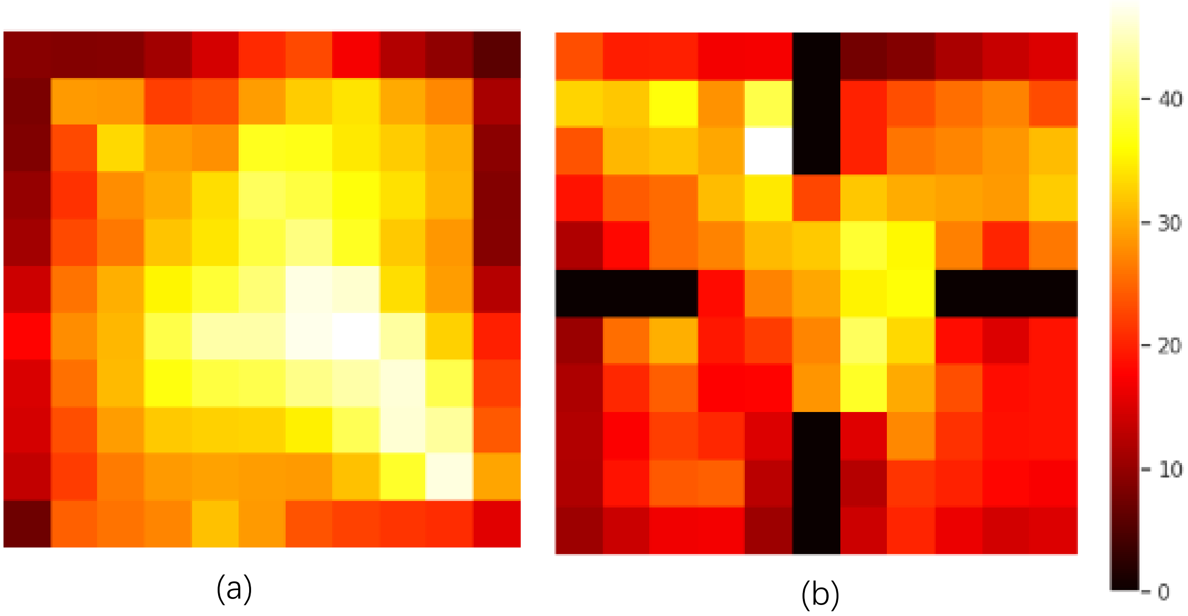


Figure 5.8: ErrP gradient distribution at the different positions of two environments.

robust to noise during the test.

5.3.2 Gradient analysis at different positions

We found that ErrP has more effects on the critical location based on the neural network weights. This is useful for human involved control, as human can concentrate more on critical area.

5.3.3 ErrP Accuracy threshold for training

We investigate the agent performance with various noise levels during training. We sweep the input accuracy during the training phase. We found that the agent learned different policies while training with different ErrP accuracy levels, demonstrating that the model could infer the partially observation level during training. Besides, we investigated the threshold of ErrP accuracy which is sufficient to train an efficient shared control model. We found that if the ErrP accuracy is higher than 70%, the model trained with this level ErrP accuracy will do better with a solo autonomous agent. However, as if the ErrP accuracy is lower than 70%, the shared control performance was not compatible with a sole autonomous agent since the sole agent can learn a search policy without human feedback. If the ErrP accuracy is poor, it will harm the performance of the task. The result provided a perspective view for the shared control critic regarding human feedback accuracy. However, the finding of the threshold was only based on the experiment, the theoretical explanation based on the neural network could be conducted in future research.

5.3.4 Adaptive human-robot interaction

As a baseline, the solo agent learned an optimal search trajectory for each possible start point without human feedback. Once the start point was fixed, the agent would move along the fixed path until the target was found. The trajectory is fixed depending on the agent's start point for solo agent without human feedback. However, the agent search path was dynamically changed during shared control to adapt the human mindset with ErrP feedback. This is an adaptive search strategy for the agent. From the agent's perspective, the user acts like a critic who evaluates the agent's action to infer the target position. From the user's perspective, the agent behaves like an adaptive interface that (inferring the goal position) learns a personalized mapping from user feedback and agent observation to actions that maximize task reward.

5.3.5 Efficiency of shared control

Even for the best search strategy, the autonomous agent still needs to explore for a long time to gain information at the initial time. With human-guided direction, the search efficiency will be greatly improved. One sensible approach to speed up the search is to focus on promising regions. Human guidance provides trial-and-error feedback to narrow its search to the most promising areas rather than exhaustively enumerating solutions [16]. Imagine the searching task in a very big maze. This problem is challenging for robots with limited horizons. If a robot's searching direction is different to the target direction, it will take a long time for the robot to realize the mistake (such as reaching the edge), which is difficult and expensive to explore with limited sensor horizons. For a robot, understanding what makes a good search direction can be difficult with limited horizons about the environment. Shared control addresses the target searching problem by combining human input with automated assistance. Humans are naturally aware of the domain knowledge of the intrinsic structure of the maze, and, consequently, they can provide feedback during robot exploration. Indeed, this feedback is a very powerful way to provide criticism while the robot is exploring a large environment maze. By establishing collaboration between a human user and a robot in exerting control over a system, a robot can utilize the user's guide during a task.

CHAPTER 6

VALIDATION THE LEARNED SHARED AUTONOMY WITH REAL HUMAN PARTICIPANTS

Following the experiment results shown in Chapter 5, this chapter will present the experiment overview of user study and the results of testing the feasibility of the shared autonomy with real EEG data from real human participants, using the model learned via deep recurrent reinforcement learning. The latter part of this chapter will discuss those results.

6.1 Experiment overview

The main purpose of this experiment is to evaluate our shared control model during the test phase with real human participants. The model we used is pre-trained with simulated EEG. We want to validate the feasibility that the model trained with simulated EEG can be used in real human participants in the same task environment, as shown in Figure 6.1.

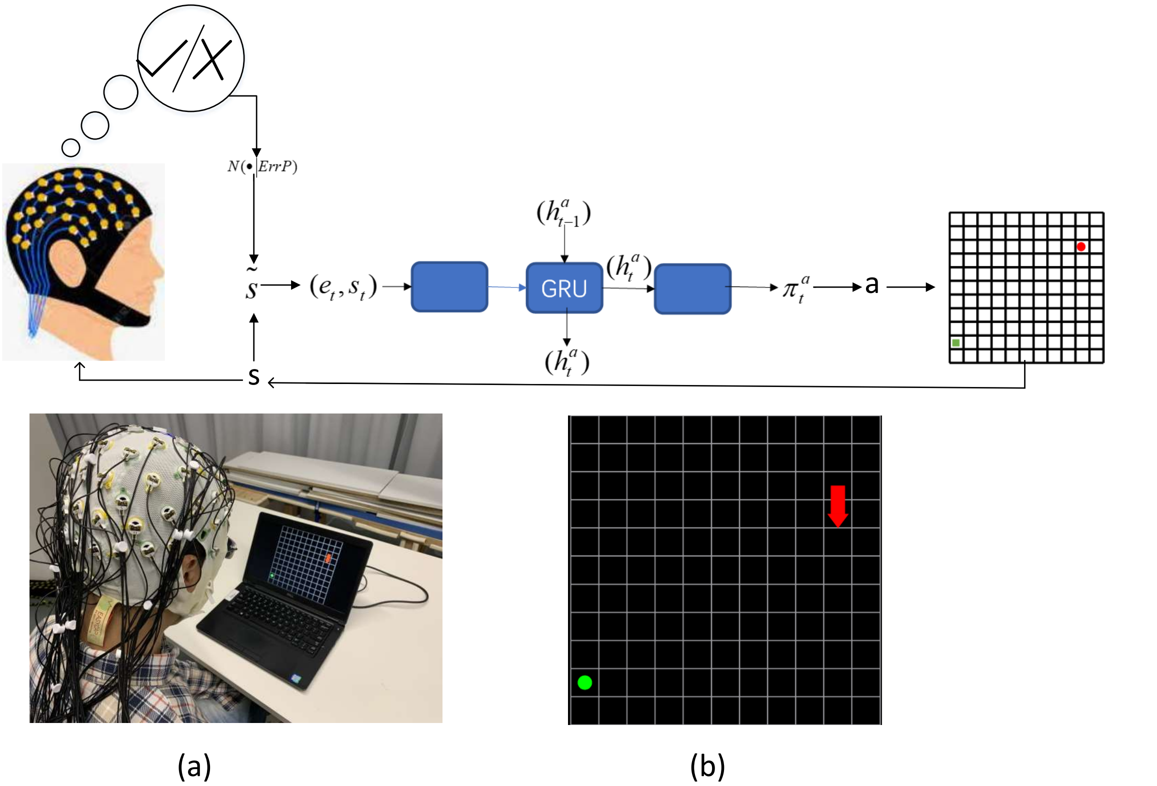
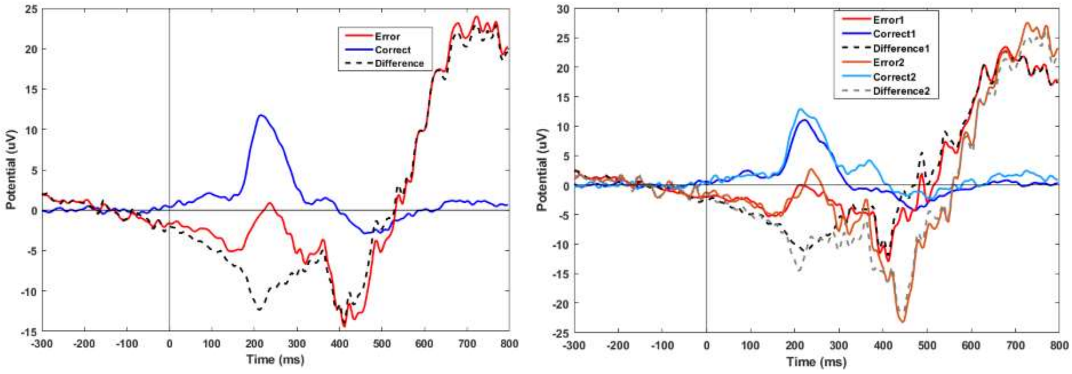


Figure 6.1: Shared autonomy Frame architecture. We evaluated our method in a target search task with real human participants (a). The red dot with an arrow is the agent , and the green square is the target (b)

6.2 Results analysis

6.2.1 Electrophysiology analysis

Figure 6.2 (a) shows the correct, error, and difference grand averaged potentials (error minus correct averages) in channel Fz, averaged for all the subjects combined two scenarios. The grand difference average was characterized by three components: a negative deflection at around 200 ms, a positive deflection at around 350 ms, and another negative component at around 400 ms. The results agree with previous studies about ErrP [77]. We also compare the ErrP of two scenarios to validate the consistency of the potentials. As shown in Figure 6.2 (b), the shape of peak and peak latency is quite



(a)

(b)

Figure 6.2: ERP analysis for correct and error conditions averaged trials of scenario 1 and scenario 2 (a). ERP analysis for correct and error conditions of scenario 1 and scenario 2, respectively (b). The legend numbers "1" and "2" refer to scenario 1 and scenario 2, respectively.

consistent across the two environments. That indicated that the ErrP classifier trained with another environment could be used in a different environment. Thus, we use the classifier trained with scenario 1 on two scenarios during the test. The application of ErrPs to reasonably complex environments demonstrated that ErrP classifiers could be transferred to another unseen environment with zero-shot learning. This proved that the ErrP decoder could be trained and tested in varied environments. This also confirms the superiority of ErrP over other paradigms like MI and SSEVP in terms of command mapping. ErrP does not need to map the command to agent performance, but only cognitive judgment, correct or error. It can be used in diverse tasks, implying that the ErrP-based paradigm is generic. This is in contrast to another paradigm that caters to a specific category. This is the advantage for the ErrP-based paradigm, in that it is not limited to a specific stimulus or modality. However, the premise that the ErrP classifier can be used in different environments is that the stimuli should be the same between environments since the potential was related to stimuli size and shape [98], which could affect the peak potentials and latency.

Participant	Offline training (%)	Scenario 1 block 1 (%)	Scenario 1 block 2 (%)	Scenario 2 block 1 (%)	Scenario 2 block 2 (%)
S2	69.50	74.5	81.92	75.51	79.34
S4	72.83	56.52	75.35	73.68	63.13
S5	85.17	96.82	92.66	84.59	46.43
S6	74.67	73.98	69.25	67.51	65.92
Average	75.54	75.46	79.80	75.32	63.71

Table 6.1: ErrP training accuracy with 10-fold cross-validation and testing accuracy for the two scenarios.

6.2.2 Classification analysis of ErrP

In this section, we analyze the real-time classification accuracy of ErrP with the classification model calibrated with offline data for the two scenarios. As mentioned in the simulation section, ErrP classification accuracy is vital for agent performance. If ErrP accuracy is lower than 70%, the performance with low ErrP classification accuracy is no more superior to no ErrP condition. Table 6.1 was the offline training accuracy using 10-fold-validation. We excluded the participants S1 and S3 due to the poor training accuracy as this will not work on the learned model. One would argue that removing the participants with poor performance would bias toward better performance. However, we just want to demonstrate the feasibility of the shared control when the ErrP accuracy is above the threshold (70%). This is also why we only analyze the ErrP accuracy.

We analyze the test ErrP accuracy on the blocks for the two scenarios, respectively. Needing to be checked here is the relationship between test ErrP accuracy and steps used to reach the target position for the agent for each block. As shown in Table 6.1, when combined with block 1 and block 2, the overall accuracy was 74.43% and 67.22% for scenario 1 and scenario 2, respectively. The lower accuracy for scenario 2 was due to a significant drop in accuracy of participant S5. As the participant was on medication at the end of the experiment.

In this section, we analyze agent performance with real human feedback. We first examined the agent's performance in the simulation environment. We compared the performance in simulation and real environments to assess the feasibility of our method with real human participants during the test phase.

Performance was operationalized as the number of steps taken by the agent to reach the target position from the start position. Even without human assistance, the agent would eventually reach the target with its observation and the learned policy.

To check the search ability of different distance ranges, we set the initial distance ranging from 2 to 20. Each distance consists of 10000 runs. Figure 6.3 sets out the averaged steps used of each initial distance between agent start position and target position for two scenarios with no ErrP, 70% accuracy ErrP, and 80% accuracy ErrP. The averaged steps were 51.2, 34.8, and 24.0 for no ErrP condition, 70% accuracy ErrP and 80% accuracy condition for scenario 1, and 50.7, 40.8, and 25.7 for scenario 2. The averaged steps were gradually increased when the initial distance expanded for both ErrP condition and no ErrP condition for scenario 1 and scenario 2. The exception was when the initial distance was maximum for no ErrP condition in scenario 2. The averaged step dropped to 49.9 for the maximum initial distance for no ErrP condition in scenario 2. As only if the start and target positions were limited to the four corner positions, the initial distance could be the maximum. We found that when the agent start position was (0,0), the agent moved in the opposite direction. If the agent start and target positions were (0,0) and (10,10), respectively, it used 20 steps to reach the target position, which represent the ideal.

Taking 60 steps as the maximum number, if the agent successfully reaches the target position within 60 steps, then it is can be deemed a success. Otherwise, it failed. Figure 6.4 summarizes the success rate for reaching the target position within 60 steps for each initial distance. The success rate was gradually decreased as the initial distance increased for 70% accuracy ErrP and no ErrP conditions for the two scenarios. The averaged success rates were 79.74%, 83.19% and 95.86% for no ErrP, 70% accuracy ErrP and 80% accuracy ErrP conditions for scenario 1. For scenario 2 they were 69.43%, 76.70%

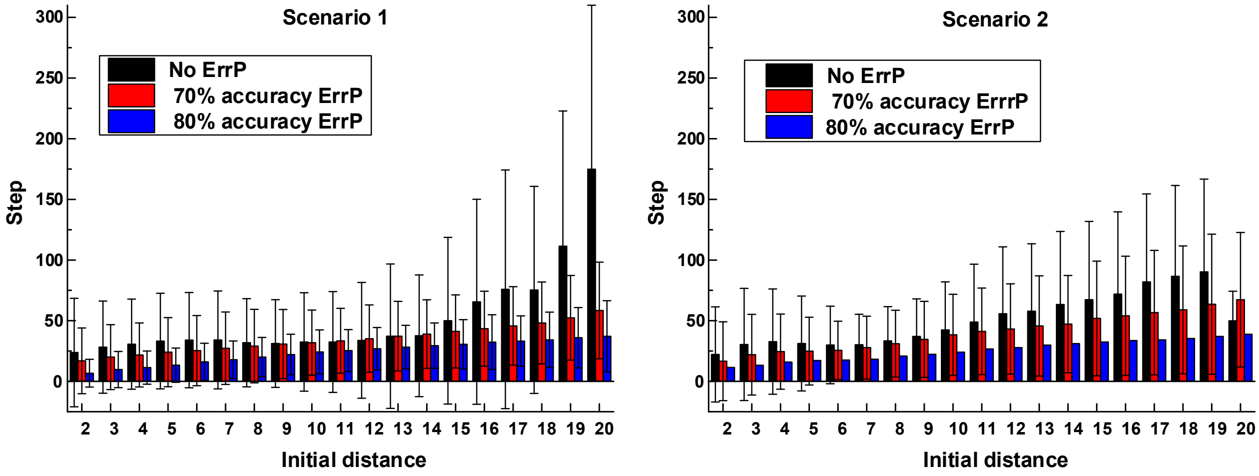


Figure 6.3: The averaged steps used among 10000 times of each initial distance between agent start position and target position for two scenarios with no ErrP, 70% accuracy ErrP, and 80% accuracy ErrP.

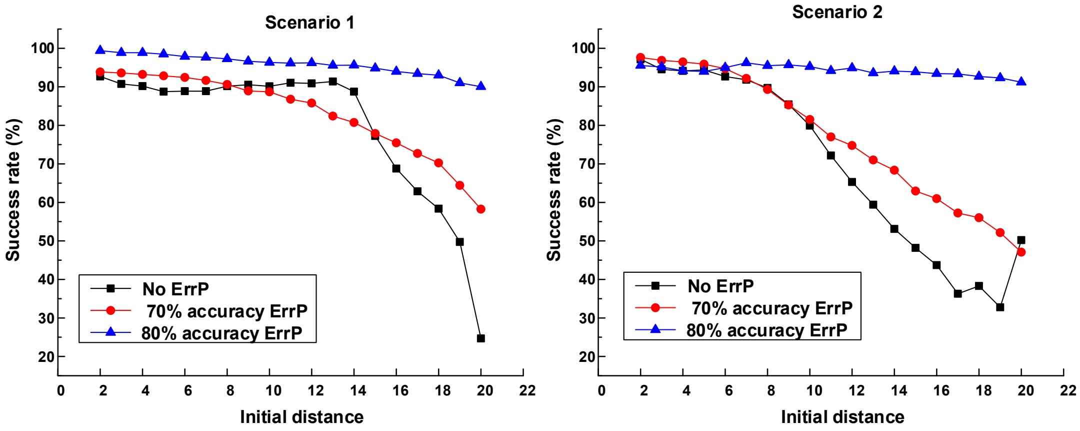


Figure 6.4: Success rate to reach the target position within 60 steps for each initial distance with no ErrP, 70% accuracy ErrP, and 80% accuracy ErrP conditions for scenario 1 and scenario 2.

and 94.21% for no ErrP and 70% accuracy ErrP and 80% accuracy ErrP conditions. The success rate was better for the 70% accuracy condition than for the no ErrP condition.

We analyzed the averaged steps by removing the trials that failed to reach the target positions within 60 steps. Figure 6.5 lists the averaged steps for no ErrP, 70% accuracy

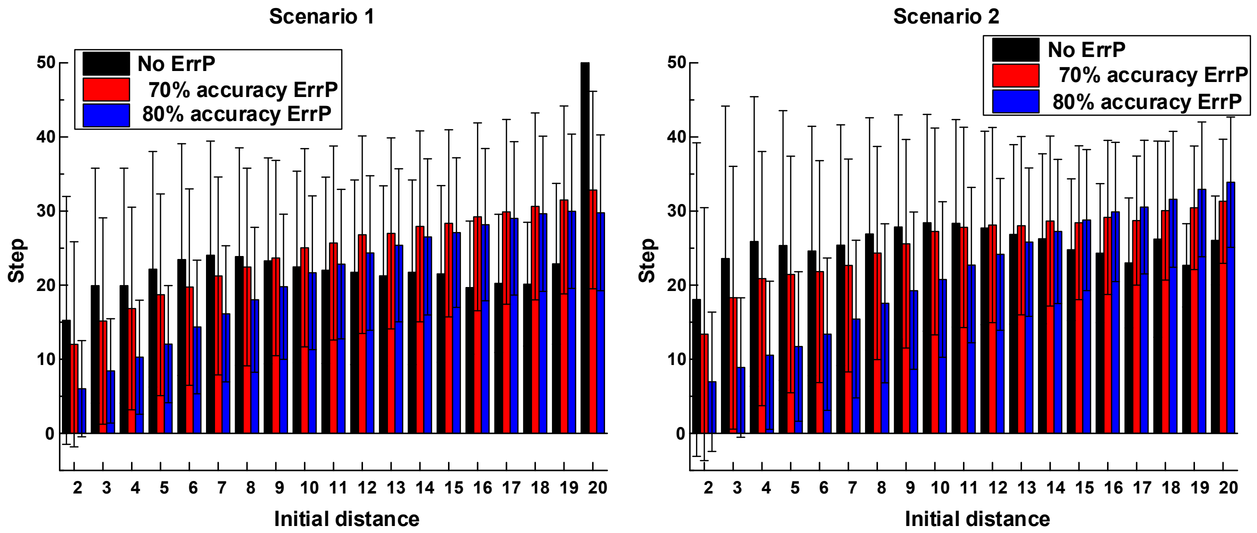


Figure 6.5: The averaged steps by removing the trials that failed to reach the target positions within 60 steps among 10000 times of each initial distance for two scenarios with no ErrP, 70% accuracy ErrP, and 80% accuracy ErrP.

ErrP, and 80% ErrP accuracy for two scenarios. The averaged steps were 28.4, 24.4 and 20.8 for no ErrP, 70% accuracy ErrP and 80% accuracy ErrP condition for scenario 1, and 25.4, 25.6 and 21.6 for no ErrP, 70% accuracy ErrP and 80% accuracy ErrP condition for scenario 2.

The simulation results confirmed that using ErrP as an agent input can significantly reduce the steps used to reach the target position compared with that of no ErrP condition. Although the ErrP accuracy was only 70% with correct probability, it still can improve the performance by more than 30%.

During the online test, the agent's last action's classification as either correct or an error was fed in real-time to the model as one of the inputs for generating next action command. If the agent did not reach the target position after 60 steps, this run would be finished and a new sample would commence. The maximum number of steps was set in order so as not to discourage the participants after a long time. The maximum steps was

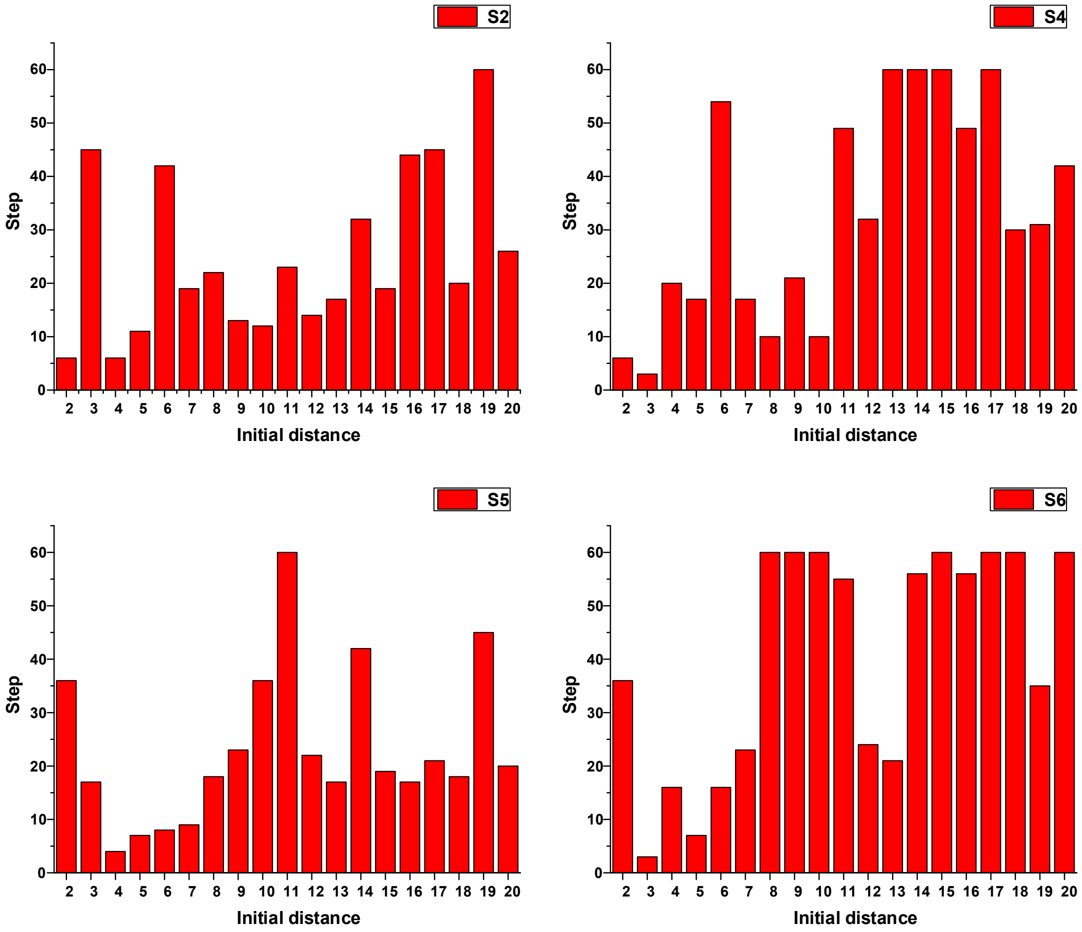


Figure 6.6: Steps used for participants in real experiment for scenario 1.

set to 60 for each sample for the real experiment. Therefore, the sample with steps that over 60 steps were included in the averaged steps.

Figure 6.6lists the steps used for the four participants for scenario 1. As shown in Table 6.2, the success rate for reaching the target position within 60 steps was 84.21%, averaged for all the participants. The average step was 24.18 averaged for all the participants by removing the failed samples. The success rate was similar to that with 70% accuracy ErrP condition 83.19% and bigger than no ErrP condition 79.74% in the simulation. The averaged step was almost the same with 24.4 of 70% accuracy ErrP and smaller than no ErrP condition at 28.4.

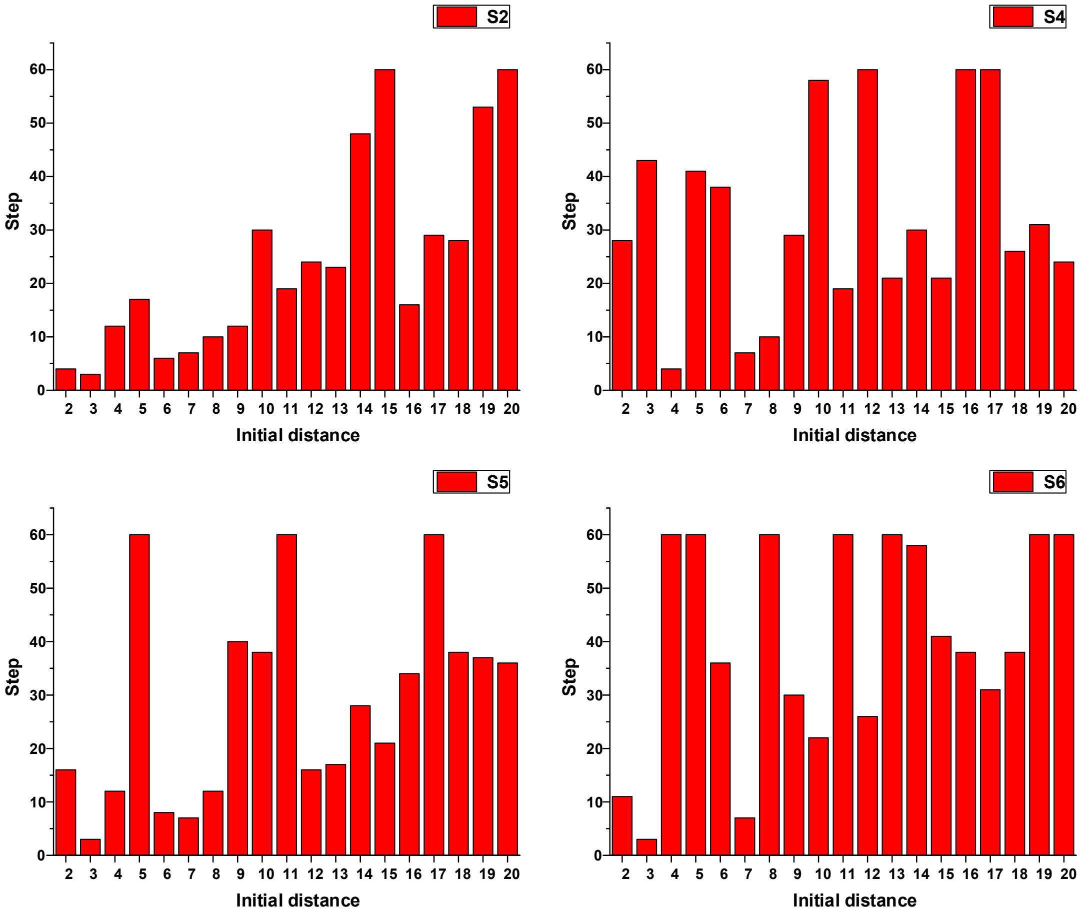


Figure 6.7: Steps used for participants in real experiment for scenario 2.

Figure 6.7 shows the steps used for the four participants in scenario 2. As shown in Table 6.2, the success rate to reach the target position within 60 steps was 80.26%, averaged for all the participants. The average step was 24.35 for all participants and achieved by removing the failed sample. The success rate was better than that with 70% accuracy ErrP condition 76.70% and no ErrP condition 69.43% in simulation. The averaged step was smaller than 70% accuracy ErrP 25.6 and no ErrP condition at 25.4.

The success rate was lower in scenario 2 than in scenario 1, both for simulated and real experiments, since the environment was more complex with obstacles in scenario 2. The obstacles divided the environment into several regions. Once the agent moves into

	Success rate (%)		Averaged steps	
	Env 1	Env 2	Env 1	Env 2
S2	94.74	89.47	23.1	20.06
S4	78.95	84.21	22.2	26.71
S5	94.74	84.21	21.2	22.44
S6	63.13	63.16	29	28.42
Ave	84.21	80.26	24.35	24.18

Table 6.2: Table: Success rate and averaged steps in real experiment.

the wrong region, it will take several steps to move out of the region. Apart from this, for a complex maze, it is difficult for a user to evaluate the agent's action whether it is a good one or not in the real experiment. This could be shown the level of ErrP accuracy. The accuracies were 75.46% and 79.80, respectively, for block 1 and block 2 for scenario 1, and 75.32 and 63.71 regarding two blocks for scenario 2.

6.3 Discussion

6.4 Interaction design

ErrP evocation needs precise locked stimulus, for example robot action or a clear visual cue. In our research, the discrete action with arrow flash was used as the stimulus. It is, however, difficult to use the robot's action as the stimulus in a real environment. Furthermore, it is unnatural to divide the robot's continuous movement into discrete actions. Thus, how to design the ErrP stimulus during human-robot interaction is a key point. One potential way to overcome this is to add an additional communication channel on the robot, like LCD, to serve as the medium for stimulus.

Feasibility of Simulated ErrP for training and real for testing: We demonstrated the feasibility of our method of training with simulated data and testing with real EEG data from human participants in real-time. The key idea was that the simulated data had a binary value (0 or 1), based on the ErrP classifier, whose output was a binary value (0 or 1). The simulated pilot enables us to train the model without real users. After all, real EEG data collections can be very time-consuming and have other drawbacks, such as overfitting if there is not enough data. We use binary values instead of a linear scale between 0 and 1 to reduce the gap between simulation data and real EEG data classification results. To train the model, one can also use simulated ErrP scaled linearly from 0 to 1. Correspondingly, the classifier's output should be scaled linearly with real ErrP data, which is related to the consistency with goals as mentioned in [192].

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusion

This thesis presents two novel ErrP-based BCI systems. we first designed two experiments in Chapter 3. In respond to research aim one, the first experiment was to explore the observability and decodability of ErrP in a continuous interaction between the human and the robot. In responding to the research aim two and three, the second experiment was to explore the feasibility of building an ErrP-based shared autonomy via deep recurrent reinforcement learning.

In Chapter 4, we presented the result and discussion of a neuro-based method for human observers that enables evaluating robot intentions in real-time via the LCD and thus allows continuous evaluation of robot's behaviors. The observability and decodability of ErrPs as well as the high classification accuracy (77.57%) in response to robot intention demonstrated the system usability. We also found that the observability and decodability of ErrP were better when the interaction sequence was close to the robot decision point among the continuous interaction sequences at Stage 1, which provided a potential time window for such early intervention in this BCI paradigm. Further experiments might be needed to examine if more sophisticated algorithms or different visual stimuli could expand the time window for early intervention. In addition, different levels of human observer engagement between online and offline sessions can significantly affect online performance. Online adaptive learning would be implemented for future research to adapt to the variation between offline and online sessions.

We presented and discussed the experiment results of the ErrP-based shared control autonomy via deep recurrent reinforcement with simulated EEG data in Chapter 5. To handle the ErrP uncertainty, we formula the learning as POMDP and use RNN to deal with POMDP. We found that shared autonomy can greatly improve the agent's performance. We also found that the model trained with partial observation was more robust to ErrP uncertainty during the test. Besides, the agent learned different policies while training with different ErrP accuracy levels. The results provided a perspective view for the shared control critic regards human feedback accuracy.

Chapter 6 validated the shared autonomy with real-time decoded EEG from real human participants on two target search environments, using the trained model as described in Chapter 5. The results showed that the agent's performance is better with human feedback than without human feedback. The agent can adaptively change its search direction with human feedback. In a simulation with 70% ErrP accuracy, agents completed the task 14.1 % faster. In the real-world experiment, agents completed the navigation task 14.9% faster. The good performance in real experiments suggests that this method can be effective at human-robot shared autonomy with uncertainty noise input, such as neuro activities, which demonstrated the feasibility of the shared autonomy in the real environment.

These two novel BCI systems advance the current ErrP-based BCI capabilities and enable a wide range of new interaction possibilities between human and robot. This thesis represents an important step toward a BCI-based shared autonomy between humans and robots.

7.2 Future work

In the shared control method, the user needs to monitor the robot's action in each step. The continuous involvement will cause a burden for the human user. This is far from optimal solutions and not friendly in human-computer interaction tasks. Intermittent feedback is a promising strategy alternative to human-continuous feedback toward every step. The human user does not need to monitor the agent's action at every step. The user can intervene when there is a need, for example, when the agent has the potential to make mistakes, which can significantly reduce human wordload. In this case, the human user doesn't need to monitor all the time. The model can react to human input at any time. The intermittent controller can be established through reinforcement learning by naturally formulating human feedback into three values: positive, negative, and no feedback. In corresponding, a classification algorithm needs to be used to detect whether the user involves the task from brain signals. The shared autonomy with intermittent human feedback can be conducted in future work.

The proposed autonomy approach is evaluated with real human participants in a 2D grid environment. Future research would extend this method with humans and real robots.

The finding of the threshold of the ErrP accuracy to train an effective model was based on experimental results. Further research could be conducted to analyze the computational neural network model to figure out the threshold.

Bibliography

[1] *LiveAmp 64 » Brain Vision*.
https://brainvision.com/products/liveamp-64/.

[2] I. Akinola, B. Chen, J. Koss, A. Patankar, J. Varley, and P. Allen, *Task level hierarchical system for BCI-enabled shared autonomy*, in 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids), IEEE, 2017, pp. 219–225.

[3] I. Akinola, Z. Wang, J. Shi, X. He, P. Lapborisuth, J. Xu, D. Watkins-Valls, P. Sajda, and P. Allen, *Accelerated robot learning via human brain signals*, in 2020 IEEE international conference on robotics and automation (ICRA), IEEE, 2020, pp. 3799–3805.

[4] H. Al-Negheimish, L. Al-Andas, L. Al-Mofeez, A. Al-Abdullatif, N. Al-Khalifa, and A. Al-Wabil, *Brainwave typing: Comparative study of p300 and motor imagery for typing using dry-electrode EEG devices*, in International Conference on Human-Computer Interaction, Springer, 2013, pp. 569–573.

[5] M. Alimardani and K. Hiraki, *Passive Brain-Computer Interfaces for Enhanced Human-Robot Interaction*, Frontiers in Robotics and AI, 7 (2020).

[6] R. C. Arkin and R. C. Arkin, *Behavior-based robotics*, MIT press, 1998.

[7] D. Arumugam, J. K. Lee, S. Saskin, and M. L. Littman, *Deep reinforcement learning from policy-dependent human feedback*, arXiv preprint arXiv:1902.04257, (2019).

[8] L. Bi, X.-A. Fan, and Y. Liu, *EEG-based brain-controlled mobile robots: A survey*, IEEE transactions on human-machine systems, 43 (2013), pp. 161–176.

[9] N. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, A. Kübler, J. Perelmouter, E. Taub, and H. Flor, *A spelling device for the paralysed*, Nature, 398 (1999), pp. 297–298.

[10] B. Blankertz, G. Dornhege, C. Schafer, R. Krepki, J. Kohlmorgen, K.-R. Muller, V. Kunzmann, F. Losch, and G. Curio, *Boosting bit rates and error detection for the classification of fast-paced motor commands based on single-trial EEG analysis*, IEEE Transactions on Neural Systems and Rehabilitation Engineering, 11 (2003), pp. 127–131.

[11] J.-M. Bollon, R. Chavarriaga, J. d. R. Millán, and P. Bessiere, *EEG error-related potentials detection with a Bayesian filter*, in 2009 4th International IEEE/EMBS Conference on Neural Engineering, IEEE, 2009, pp. 702–705.

[12] J. M. Bradshaw, A. Acquisti, J. Allen, M. R. Breedy, L. Bunch, N. Chambers, P. Feltovich, L. Galescu, M. A. Goodrich, and R. Jeffers, *Teamwork-centered autonomy for extended human-agent interaction in space applications*, in AAAI 2004 Spring Symposium, 2004, pp. 22–24.

[13] C. Breazeal, *Emotion and sociable humanoid robots*, International journal of human-computer studies, 59 (2003), pp. 119–155.

[14] F. Broz, I. Nourbakhsh, and R. Simmons, *Designing pomdp models of socially situated tasks*, in 2011 RO-MAN, IEEE, 2011, pp. 39–46.

[15] A. BRUCE, J. KNIGHT, S. LISTOPAD, B. MAGERKO, AND I. R. NOURBAKHSH, *Robot improv: Using drama to create believable agents*, in Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), vol. 4, IEEE, 2000, pp. 4002–4008.

[16] L. BURKS, N. AHMED, I. LOEFGREN, L. BARBIER, J. MUESING, J. MCGINLEY, AND S. VUNNAM, *Collaborative human-autonomy semantic sensing through structured POMDP planning*, Robotics and Autonomous Systems, 140 (2021), p. 103753.

[17] A. BUTTFIELD, P. W. FERREZ, AND J. R. MILLAN, *Towards a robust BCI: error potentials and online learning*, IEEE Transactions on Neural Systems and Rehabilitation Engineering, 14 (2006), pp. 164–168.

[18] T. CARLSON AND J. D. R. MILLAN, *Brain-controlled wheelchairs: a robotic architecture*, IEEE Robotics & Automation Magazine, 20 (2013), pp. 65–73.

[19] L. CARRETIÉ, F. MERCADO, M. TAPIA, AND J. A. HINOJOSA, *Emotion, attention, and the 'negativity bias', studied through event-related potentials*, International journal of psychophysiology, 41 (2001), pp. 75–85.

[20] J. F. CAVANAGH AND M. J. FRANK, *Frontal theta as a mechanism for cognitive control*, Trends in cognitive sciences, 18 (2014), pp. 414–421.

[21] R. CHAVARRIAGA AND J. D. R. MILLÁN, *Learning from EEG error-related potentials in noninvasive brain-computer interfaces*, IEEE transactions on neural systems and rehabilitation engineering, 18 (2010), pp. 381–388.

[22] R. CHAVARRIAGA, X. PERRIN, R. SIEGWART, AND J. D. R. MILLÁN, *Anticipation- and error-related EEG signals during realistic human-machine interaction: A*

study on visual and tactile feedback, in 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Ieee, 2012, pp. 6723–6726.

[23] R. Chavarriaga, A. Sobolewski, and J. d. R. Millán, *Errare machinale est: the use of error-related potentials in brain-machine interfaces*, Frontiers in neuroscience, 8 (2014), p. 208.

[24] X. Chen, B. Zhao, Y. Wang, S. Xu, and X. Gao, *Control of a 7-DOF robotic arm system with an SSVEP-based BCI*, International journal of neural systems, 28 (2018), p. 1850018.

[25] K. Choi and A. Cichocki, *Control of a wheelchair by motor imagery in real time*, in International conference on intelligent data engineering and automated learning, Springer, 2008, pp. 330–337.

[26] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, *Deep reinforcement learning from human preferences*, arXiv preprint arXiv:1706.03741, (2017).

[27] W. J. Clancey, *Field science ethnography: Methods for systematic observation on an Arctic expedition*, Field Methods, 13 (2001), pp. 223–243.

[28] E. Colgate, A. Bicchi, M. A. Peshkin, and J. E. Colgate, *Safety for physical human-robot interaction*, in Springer handbook of robotics, Springer, 2008, pp. 1335–1348.

[29] C. Cortes and V. Vapnik, *Support-vector networks*, Machine learning, 20 (1995), pp. 273–297.

[30] B. DAL SENO, M. MATTEUCCI, AND L. MAINARDI, *Online detection of P300 and error potentials in a BCI speller*, Computational intelligence and neuroscience, 2010 (2010).

[31] T. DEGRIS, P. M. PILARSKI, AND R. S. SUTTON, *Model-Free reinforcement learning with continuous action in practice*, in 2012 American Control Conference (ACC), June 2012, pp. 2177–2182.

[32] C. L. DIAS, A. I. SBURLEA, AND G. R. MÜLLER-PUTZ, *Masked and unmasked error-related potentials during continuous control and feedback*, Journal of neural engineering, 15 (2018), p. 036031.

[33] E. W. DIJKSTRA, *A note on two problems in connexion with graphs*, Numerische mathematik, 1 (1959), pp. 269–271.

[34] N. R. DRAPER AND H. SMITH, *Applied regression analysis*, vol. 326, John Wiley & Sons, 1998.

[35] S. EHRLICH AND G. CHENG, *A neuro-based method for detecting context-dependent erroneous robot action*, in 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), IEEE, 2016, pp. 477–482.

[36] S. K. EHRLICH AND G. CHENG, *Human-agent co-adaptation using error-related potentials*, Journal of neural engineering, 15 (2018), p. 066014.

[37] S. K. EHRLICH AND G. CHENG, *A feasibility study for validating robot actions using eeg-based error-related potentials*, International Journal of Social Robotics, 11 (2019), pp. 271–283.

[38] S. EL ZAATARI, M. MAREI, W. LI, AND Z. USMAN, *Cobot programming for collaborative industrial tasks: An overview*, Robotics and Autonomous Systems, 116 (2019), pp. 162–180.

[39] M. R. ENDSLEY, B. BOLTÉ, AND D. G. JONES, *Designing for situation awareness: An approach to user-centered design*, CRC press, 2003.

[40] M. S. ERDEN AND B. MARIĆ, *Assisting manual welding with robot*, Robotics and Computer-Integrated Manufacturing, 27 (2011), pp. 818–828.

[41] M. FALKENSTEIN, *Effects of errors in choice reaction tasks on the ERP under focused and divided attention*, Psychophysiological brain research, (1990).

[42] M. FALKENSTEIN, J. HOORMANN, S. CHRIST, AND J. HOHNSBEIN, *ERP components on reaction errors and their functional significance: a tutorial*, Biological psychology, 51 (2000), pp. 87–107.

[43] J. FALLER, *BCIs that use steady-state visual evoked potentials or slow cortical potentials*, in Brain-Computer Interfaces, Principles and Practice, Oxford University Press, 2012.

[44] L. A. FARWELL AND E. DONCHIN, *Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials*, Electroencephalography and clinical Neurophysiology, 70 (1988), pp. 510–523.

[45] F. C. FERNANDEZ AND W. CAARLS, *Deep Reinforcement Learning for Haptic Shared Control in Unknown Tasks*, arXiv preprint arXiv:2101.06227, (2021).

[46] P. W. FERREZ AND J. D. R. MILLÁN, *You Are Wrong!—Automatic Detection of Interaction Errors from Brain Waves*, in Proceedings of the 19th international joint conference on artificial intelligence, 2005.

[47] P. W. FERREZ AND J. D. R. MILLAN, *Error-related EEG potentials generated during simulated brain–computer interaction*, IEEE transactions on biomedical engineering, 55 (2008), pp. 923–929.

[48] P. W. Ferrez and J. d. R. Millán, *Simultaneous real-time detection of motor imagery and error-related potentials for improved BCI accuracy*, in Proceedings of the 4th international brain-computer interface workshop and training course, 2008, pp. 197–202.

[49] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, *Counterfactual multi-agent policy gradients*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, 2018.

[50] T. Fong and C. Thorpe, *Vehicle teleoperation interfaces*, Autonomous robots, 11 (2001), pp. 9–18.

[51] G. Foresi, A. Freddi, S. Iarlori, A. Monteriu, D. Ortenzi, and D. P. Pagnotta, *Human-robot cooperation via brain computer interface*, in 2017 IEEE 7th International Conference on Consumer Electronics-Berlin (ICCE-Berlin), IEEE, 2017, pp. 1–2.

[52] J. Forlizzi and C. DiSalvo, *Service robots in the domestic environment: a study of the roomba vacuum in the home*, in Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction, 2006, pp. 258–265.

[53] V. Gandhi, G. Prasad, D. Coyle, L. Behera, and T. M. McGinnity, *EEG-based mobile robot control through an adaptive brain–robot interface*, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 44 (2014), pp. 1278–1285.

[54] G. Garakani, H. Ghane, and M. B. Menhaj, *Control of a 2-DoF robotic arm using a P300-based brain-computer interface*, arXiv preprint arXiv:1901.01422, (2019).

[55] W. J. GEHRING, B. GOSS, M. G. COLES, D. E. MEYER, AND E. DONCHIN, *A neural system for error detection and compensation*, Psychological science, 4 (1993), pp. 385–390.

[56] T. GENG AND J. Q. GAN, *Motor prediction in brain-computer interfaces for controlling mobile robots*, in 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, 2008, pp. 634–637.

[57] T. GENG, J. Q. GAN, AND H. HU, *A self-paced online BCI for mobile robot control*, International Journal of Advanced Mechatronic Systems, 2 (2010), pp. 28–35.

[58] R. GOCKLEY, A. BRUCE, J. FORLIZZI, M. MICHALOWSKI, A. MUNDELL, S. ROSENTHAL, B. SELLNER, R. SIMMONS, K. SNIPES, AND A. C. SCHULTZ, *Designing robots for long-term social interaction*, in 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2005, pp. 1338–1343.

[59] M. K. GOEL, R. CHAVARRIAGA, AND J. D. R. MILLÁN, *Cortical current density vs. surface EEG for event-related potential-based Brain-Computer interface*, in 2011 5th International IEEE/EMBS Conference on Neural Engineering, IEEE, 2011, pp. 430–433.

[60] M. A. GOODRICH AND A. C. SCHULTZ, *Human-robot interaction: a survey*, Now Publishers Inc, 2008.

[61] N. GOPALAN AND S. TELLEX, *Modeling and solving human-robot collaborative tasks using pomdps*, in RSS Workshop on Model Learning for Human-Robot Communication, vol. 32, 2015, pp. 590–628.

[62] E. GREENSMITH, P. L. BARTLETT, AND J. BAXTER, *Variance Reduction Techniques for Gradient Estimates in Reinforcement Learning.*, Journal of Machine Learning Research, 5 (2004).

[63] J. GRIZOU, I. ITURRATE, L. MONTESANO, P.-Y. OUDEYER, AND M. LOPES, *Calibration-free BCI based control*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28, 2014.

[64] C. GUGER, W. HARKAM, C. HERTNAES, AND G. PFURTSCHELLER, *Prosthetic control by an EEG-based brain-computer interface (BCI)*, in Proc. aaate 5th european conference for the advancement of assistive technology, Citeseer, 1999, pp. 3–6.

[65] A. GÜNEYSU AND H. L. AKIN, *An SSVEP based BCI to control a humanoid robot by using portable EEG device*, in 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, 2013, pp. 6905–6908.

[66] T. HASTIE, R. TIBSHIRANI, AND J. FRIEDMAN, *The elements of statistical learning: data mining, inference, and prediction*, Springer Science & Business Media, 2009.

[67] S. HAUFE, F. MEINECKE, K. GÖRGEN, S. DÄHNE, J.-D. HAYNES, B. BLANKERTZ, AND F. BIESSMANN, *On the interpretation of weight vectors of linear models in multivariate neuroimaging*, Neuroimage, 87 (2014), pp. 96–110.

[68] M. HAUSKNECHT AND P. STONE, *Deep recurrent q-learning for partially observable mdps*, in 2015 aaai fall symposium series, 2015.

[69] M. A. HEARST, J. ALLEN, C. GUINN, AND E. HORVITZ, *Mixed-initiative interaction: Trends and controversies*, IEEE Intelligent Systems, 14 (1999), pp. 14–23.

[70] J. C. HENRY, *Electroencephalography: basic principles, clinical applications, and related fields*, Neurology, 67 (2006), pp. 2092–2092.

[71] S. HOCHREITER AND J. SCHMIDHUBER, *Long short-term memory*, Neural computation, 9 (1997), pp. 1735–1780.

[72] C. B. HOLROYD AND M. G. COLES, *The neural basis of human error processing: reinforcement learning, dopamine, and the error-related negativity.*, Psychological review, 109 (2002), p. 679.

[73] A. HOWES, X. CHEN, A. ACHARYA, AND R. L. LEWIS, *Interaction as an emergent property of a Partially Observable Markov Decision Process*, Computational interaction, (2018), pp. 287–310.

[74] C.-M. HUANG, M. CAKMAK, AND B. MUTLU, *Adaptive Coordination Strategies for Human-Robot Handovers.*, in Robotics: science and systems, vol. 11, Rome, Italy, 2015.

[75] H.-M. HUANG, *Autonomy levels for unmanned systems (ALFUS) framework volume I: Terminology version 2.0*, (2004).

[76] H.-J. HWANG, S. KIM, S. CHOI, AND C.-H. IM, *EEG-based brain-computer interfaces: a thorough literature survey*, International Journal of Human-Computer Interaction, 29 (2013), pp. 814–826.

[77] I.ITURRATE, L. MONTESANO, AND J. MINGUEZ, *Task-dependent signal variations in EEG error-related potentials for brain–computer interfaces*, Journal of neural engineering, 10 (2013), p. 026024.

[78] R. İNCE, S. S. ADANIR, AND F. SEVMEZ, *The inventor of electroencephalography (EEG): Hans Berger (1873–1941)*, Child's Nervous System, 37 (2021), pp. 2723–2724.

[79] I. Iturrate, J. M. Antelis, A. Kubler, and J. Minguez, *A noninvasive brain-actuated wheelchair based on a P300 neurophysiological protocol and automated navigation*, IEEE transactions on robotics, 25 (2009), pp. 614–627.

[80] I. Iturrate, R. Chavarriaga, L. Montesano, J. Minguez, and J. d. R. Millán, *Latency correction of error-related potentials reduces BCI calibration time*, in 6th Brain-Computer Interface Conference 2014, 2014.

[81] I. Iturrate, R. Chavarriaga, L. Montesano, J. Minguez, and J. d. R. Millán, *Teaching brain-machine interfaces as an alternative paradigm to neuroprosthetics control*, Scientific reports, 5 (2015), pp. 1–10.

[82] I. Iturrate, L. Montesano, and J.Minguez, *Shared-control brain-computer interface for a two dimensional reaching task using EEG error-related potentials*, in 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, 2013, pp. 5258–5262.

[83] I. Iturrate, L. Montesano, and J. Minguez, *Single trial recognition of error-related potentials during observation of robot operation*, in 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, IEEE, 2010, pp. 4181–4184.

[84] O. Ivlev, C. Martens, and A. Graeser, *Rehabilitation robots FRIEND-I and FRIEND-II with the dexterous lightweight manipulator*, Technology and Disability, 17 (2005), pp. 111–123.

[85] F. Iwane, M. S. Halvagal, I. Iturrate, I. Batzianoulis, R. Chavarriaga, A. Billard, and J. d. R. Millán, *Inferring subjective preferences on robot trajectories using EEG signals*, in 2019 9th International IEEE/EMBS Conference on Neural Engineering (NER), IEEE, 2019, pp. 255–258.

[86] A. JACOFF, E. MESSINA, AND J. EVANS, *A standard test course for urban search and rescue robots*, NIST special publication SP, (2001), pp. 253–259.

[87] H. JASPER, *Report of the committee on methods of clinical examination in electroencephalography*, Electroencephalogr Clin Neurophysiol, 10 (1958), pp. 370–375.

[88] S. JAVDANI, S. S. SRINIVASA, AND J. A. BAGNELL, *Shared autonomy via hindsight optimization*, Robotics science and systems: online proceedings, 2015 (2015).

[89] E. M. JEAN-BAPTISTE, P. ROTSHTEIN, AND M. RUSSELL, *POMDP based action planning and human error detection*, in IFIP International Conference on Artificial Intelligence Applications and Innovations, Springer, 2015, pp. 250–265.

[90] Z. JI, Q. LIU, W. XU, Z. LIU, B. YAO, B. XIONG, AND Z. ZHOU, *Towards Shared Autonomy Framework for Human-Aware Motion Planning in Industrial Human-Robot Collaboration*, in 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), IEEE, 2020, pp. 411–417.

[91] M. W. KADOUS, R. K.-M. SHEH, AND C. SAMMUT, *Effective user interface design for rescue robotics*, in Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction, 2006, pp. 250–257.

[92] L. P. KAELBLING, M. L. LITTMAN, AND A. R. CASSANDRA, *Planning and acting in partially observable stochastic domains*, Artificial intelligence, 101 (1998), pp. 99–134.

[93] A.-B. KARAMI, L. JEANPIERRE, AND A.-I. MOUADDIB, *Partially observable markov decision process for managing robot collaboration with human*, in 2009 21st IEEE International Conference on Tools with Artificial Intelligence, IEEE, 2009, pp. 518–521.

[94] ——, *Human-robot collaboration for a shared mission*, in 2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI), IEEE, 2010, pp. 155–156.

[95] M. J. KHAN AND K.-S. HONG, *Hybrid EEG–fNIRS-based eight-command decoding for BCI: application to quadcopter control*, Frontiers in neurorobotics, 11 (2017), p. 6.

[96] A. KILICARSLAN, S. PRASAD, R. G. GROSSMAN, AND J. L. CONTRERAS-VIDAL, *High accuracy decoding of user intentions using EEG to control a lower-body exoskeleton*, in 2013 35th annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, 2013, pp. 5606–5609.

[97] B. H. KIM, M. KIM, AND S. JO, *Quadcopter flight control using a low-cost hybrid interface with EEG-based classification and eye tracking*, Computers in biology and medicine, 51 (2014), pp. 82–92.

[98] S. K. KIM, E. A. KIRCHNER, A. STEFES, AND F. KIRCHNER, *Intrinsic interactive reinforcement learning–Using error-related potentials for real world human-robot interaction*, Scientific reports, 7 (2017), pp. 1–16.

[99] V. R. KONDA AND J. N. TSITSIKLIS, *Actor-critic algorithms*, in Advances in neural information processing systems, 2000, pp. 1008–1014.

[100] A. KREILINGER, H. HIEBEL, AND G. R. MÜLLER-PUTZ, *Single versus multiple events error potential detection in a BCI-controlled car game with continuous and discrete feedback*, IEEE transactions on biomedical engineering, 63 (2015), pp. 519–529.

[101] A. KREILINGER, C. NEUPER, AND G. R. MÜLLER-PUTZ, *Error potential detection during continuous movement of an artificial arm controlled by brain–computer*

interface, Medical & biological engineering & computing, 50 (2012), pp. 223–230.

[102] A. Kreilinger, C. Neuper, G. Pfurtscheller, and G. R. Müller-Putz, *Implementation of error detection into the graz-brain-computer interface, the interaction error potential*, in Assistive Technology from Adapted Equipment to Inclusive Environments, IOS Press, 2009, pp. 195–199.

[103] O. E. Krigolson and C. B. Holroyd, *Evidence for hierarchical error processing in the human brain*, Neuroscience, 137 (2006), pp. 13–17.

[104] O. E. Krigolson and C. B. Holroyd, *Hierarchical error processing: different errors, different systems*, Brain research, 1155 (2007), pp. 70–80.

[105] V. Kulyukin, C. Gharpure, J. Nicholson, and G. Osborne, *Robot-assisted wayfinding for the visually impaired in structured indoor environments*, Autonomous robots, 21 (2006), pp. 29–41.

[106] C.-P. Lam and S. S. Sastry, *A POMDP framework for human-in-the-loop system*, in 53rd IEEE Conference on Decision and Control, IEEE, 2014, pp. 6031–6036.

[107] O. Ledoit and M. Wolf, *Honey, I shrunk the sample covariance matrix*, The Journal of Portfolio Management, 30 (2004), pp. 110–119.

[108] M. Lehne, K. Ihme, A.-M. Brouwer, J. B. van Erp, and T. O. Zander, *Error-related EEG patterns during tactile human-machine interaction*, in 2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops, IEEE, 2009, pp. 1–9.

[109] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling, *Learning policies for partially observable environments: Scaling up*, in Machine Learning Proceedings 1995, Elsevier, 1995, pp. 362–370.

[110] J. LONG, Y. LI, H. WANG, T. YU, J. PAN, AND F. LI, *A hybrid brain computer interface to control the direction and speed of a simulated or real wheelchair*, IEEE Transactions on Neural Systems and Rehabilitation Engineering, 20 (2012), pp. 720–729.

[111] C. LOPES-DIAS, A. I. SBURLEA, K. BREITEGGER, D. WYSS, H. DRESCHER, R. WILDBURGER, AND G. R. MÜLLER-PUTZ, *Online asynchronous detection of error-related potentials in participants with a spinal cord injury using a generic classifier*, Journal of Neural Engineering, 18 (2021), p. 046022.

[112] C. LOPES-DIAS, A. I. SBURLEA, AND G. R. MÜLLER-PUTZ, *Online asynchronous decoding of error-related potentials during the continuous control of a robot*, Scientific reports, 9 (2019), pp. 1–9.

[113] F. LOTTE, *A tutorial on EEG signal-processing techniques for mental-state recognition in brain–computer interfaces*, Guide to brain-computer music interfacing, (2014), pp. 133–161.

[114] F. LOTTE, L. BOUGRAIN, A. CICHOCKI, M. CLERC, M. CONGEDO, A. RAKOTOMAMONJY, AND F. YGER, *A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update*, Journal of neural engineering, 15 (2018), p. 031005.

[115] G. MAEDA, A. MALOO, M. EWERTON, R. LIOUTIKOV, AND J. PETERS, *Anticipative interaction primitives for human-robot collaboration*, in 2016 AAAI Fall Symposium Series, 2016.

[116] C. MANDEL, T. LÜTH, T. LAUE, T. RÖFER, A. GRÄSER, AND B. KRIEG-BRÜCKNER, *Navigating a smart wheelchair with a brain-computer interface interpreting steady-state visual evoked potentials*, in 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2009, pp. 1118–1125.

[117] A. J. McDaid, S. Xing, and S. Q. Xie, *Brain controlled robotic exoskeleton for neurorehabilitation*, in 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, IEEE, 2013, pp. 1039–1044.

[118] J. Meng, S. Zhang, A. Bekyo, J. Olsoe, B. Baxter, and B. He, *Noninvasive electroencephalogram based control of a robotic arm for reach and grasp tasks*, Scientific Reports, 6 (2016), pp. 1–15.

[119] J. R. Millan, F. Renkens, J. Mourino, and W. Gerstner, *Noninvasive brain-actuated control of a mobile robot by human EEG*, IEEE Transactions on biomedical Engineering, 51 (2004), pp. 1026–1033.

[120] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, and K. Kavukcuoglu, *Learning to navigate in complex environments*, arXiv preprint arXiv:1611.03673, (2016).

[121] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, and G. Ostrovski, *Human-level control through deep reinforcement learning*, nature, 518 (2015), pp. 529–533.

[122] L. Mondada, M. E. Karim, and F. Mondada, *Electroencephalography as implicit communication channel for proximal interaction between humans and robot swarms*, Swarm Intelligence, 10 (2016), pp. 247–265.

[123] K. Muelling, A. Venkatraman, J.-S. Valois, J. E. Downey, J. Weiss, S. Javdani, M. Hebert, A. B. Schwartz, J. L. Collinger, and J. A. Bagnell, *Autonomy infused teleoperation with application to brain computer interface controlled manipulation*, Autonomous Robots, 41 (2017), pp. 1401–1422.

[124] G. R. MULLER-PUTZ, R. SCHERER, C. NEUPER, AND G. PFURTSCHELLER, *Steady-state somatosensory evoked potentials: suitable brain signals for brain-computer interfaces?*, IEEE transactions on neural systems and rehabilitation engineering, 14 (2006), pp. 30–37.

[125] R. MURPHY, J. CASPER, M. MICIRE, AND J. HYAMS, *Assessment of the NIST standard test bed for urban search and rescue*, NIST SPECIAL PUBLICATION SP, (2001), pp. 260–266.

[126] J. NAGI, H. NGO, L. M. GAMBARDELLA, AND G. A. DI CARO, *Wisdom of the swarm for cooperative decision-making in human-swarm interaction*, in 2015 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2015, pp. 1802–1808.

[127] Y. NAME, *ROBOTIS e-Manual*. https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/.

[128] C. H. NGUYEN, G. K. KARAVAS, AND P. ARTEMIADIS, *Adaptive multi-degree of freedom Brain Computer Interface using online feedback: Towards novel methods and metrics of mutual adaptation between humans and machines for BCI*, PloS one, 14 (2019), p. e0212620.

[129] M. N. NICOLESCU AND M. J. MATARIC, *Learning and interacting in human-robot domains*, IEEE Transactions on Systems, man, and Cybernetics-part A: Systems and Humans, 31 (2001), pp. 419–430.

[130] S. NIKOLAIDIS, R. RAMAKRISHNAN, K. GU, AND J. SHAH, *Efficient model learning from joint-action demonstrations for human-robot collaborative tasks*, in 2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI), IEEE, 2015, pp. 189–196.

[131] T. Noda, N. Sugimoto, J. Furukawa, M.-a. Sato, S.-H. Hyon, and J. Morimoto, *Brain-controlled exoskeleton robot for BMI rehabilitation*, in 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012), IEEE, 2012, pp. 21–27.

[132] D. Nurseitov, A. Serekov, A. Shintemirov, and B. Abibullaev, *Design and evaluation of a P300-ERP based BCI system for real-time control of a mobile robot*, in 2017 5th International Winter Conference on Brain-Computer Interface (BCI), IEEE, 2017, pp. 115–120.

[133] J. Omedes, I. Iturrate, R. Chavarriaga, and L. Montesano, *Asynchronous decoding of error potentials during the monitoring of a reaching task*, in 2015 IEEE international conference on systems, man, and cybernetics, IEEE, 2015, pp. 3116–3121.

[134] J. Omedes, I. Iturrate, J. Minguez, and L. Montesano, *Analysis and asynchronous detection of gradually unfolding errors during monitoring tasks*, Journal of neural engineering, 12 (2015), p. 056001.

[135] J. Omedes, I. Iturrate, L. Montesano, and J. Minguez, *Using frequency-domain features for the generalization of EEG error-related potentials among different tasks*, in 2013 35th annual international conference of the IEEE engineering in medicine and biology society (EMBC), IEEE, 2013, pp. 5263–5266.

[136] M. Palankar, K. J. De Laurentis, R. Alqasemi, E. Veras, R. Dubey, Y. Arbel, and E. Donchin, *Control of a 9-DoF wheelchair-mounted robotic arm system using a P300 brain computer interface: Initial experiments*, in 2008 IEEE International Conference on Robotics and Biomimetics, IEEE, 2009, pp. 348–353.

[137] Y. V. Pant, B. T. Kumaravel, A. Shah, E. Kraemer, M. Vazquez-Chanlatte, K. Kulkarni, B. Hartmann, and S. A. Seshia, *Model-based Formalization of the Autonomy-to-Human Perception Hand-off*, (2021).

[138] L. C. Parra, C. D. Spence, A. D. Gerson, and P. Sajda, *Response error correction-a demonstration of improved human-machine performance using real-time EEG monitoring*, IEEE transactions on neural systems and rehabilitation engineering, 11 (2003), pp. 173–177.

[139] D. Petit, P. Gergondet, A. Cherubini, M. Meilland, A. I. Comport, and A. Kheddar, *Navigation assistance for a bci-controlled humanoid robot*, in The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent, IEEE, 2014, pp. 246–251.

[140] R. T. Pivik, R. J. Broughton, R. Coppola, R. J. Davidson, N. Fox, and M. R. Nuwer, *Guidelines for the recording and quantitative analysis of electroencephalographic activity in research contexts*, Psychophysiology, 30 (1993), pp. 547–558.

[141] D. Puanhvuan and Y. Wongsawat, *Semi-automatic P300-based brain-controlled wheelchair*, in 2012 ICME International Conference on Complex Medical Engineering (CME), IEEE, 2012, pp. 455–460.

[142] B. Rebsamen, E. Burdet, C. Guan, H. Zhang, C. L. Teo, Q. Zeng, M. Ang, and C. Laugier, *A brain-controlled wheelchair based on P300 and path guidance*, in The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006., IEEE, 2006, pp. 1101–1106.

[143] S. Reddy, A. D. Dragan, and S. Levine, *Shared autonomy via deep reinforcement learning*, arXiv preprint arXiv:1802.01744, (2018).

[144] U. ROBOTS, *UR10 Cobots offer aging workforce solution and reduce relief worker costs for global car manufacturer*, 2017.

[145] E. ROGERS, R. R. MURPHY, AND B. ERICSON, *Agent-based expert assistance for visual problem solving*, in Proceedings of the first international conference on Autonomous agents, 1997, pp. 156–163.

[146] D. K. ROY, *Learning visually grounded words and syntax for a scene description task*, Computer speech & language, 16 (2002), pp. 353–385.

[147] N. ROY, G. BALTUS, D. FOX, F. GEMPERLE, J. GOETZ, T. HIRSCH, D. MARGARITIS, M. MONTEMERLO, J. PINEAU, AND J. SCHULTE, *Towards personal service robots for the elderly*, in Workshop on Interactive Robots and Entertainment (WIRE 2000), vol. 25, 2000, p. 184.

[148] A. F. SALAZAR-GOMEZ, J. DELPRETO, S. GIL, F. H. GUENTHER, AND D. RUS, *Correcting robot mistakes in real time using EEG signals*, in 2017 IEEE international conference on robotics and automation (ICRA), IEEE, 2017, pp. 6570–6577.

[149] L. SALEH, P. CHEVREL, F. CLAVEAU, J.-F. LAFAY, AND F. MARS, *Shared steering control between a driver and an automation: Stability in the presence of driver behavior uncertainty*, IEEE Transactions on Intelligent Transportation Systems, 14 (2013), pp. 974–983.

[150] S. SANEI AND J. A. CHAMBERS, *EEG signal processing*, John Wiley & Sons, 2013.

[151] T. SASAKI, D. BRSCIC, AND H. HASHIMOTO, *Human-observation-based extraction of path patterns for mobile robot navigation*, IEEE Transactions on Industrial Electronics, 57 (2009), pp. 1401–1410.

[152] A. R. SATTI, D. COYLE, AND G. PRASAD, *Self-paced brain-controlled wheelchair methodology with shared and automated assistive control*, in 2011 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB), IEEE, 2011, pp. 1–8.

[153] L. SCHIATTI, J. TESSADORI, N. DESHPANDE, G. BARRESI, L. C. KING, AND L. S. MATTOS, *Human in the loop of robot learning: Eeg-based reward signal for target identification and reaching task*, in 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 4473–4480.

[154] B. SCHMIDT AND L. WANG, *Depth camera based collision avoidance via active robot control*, Journal of manufacturing systems, 33 (2014), pp. 711–718.

[155] J. SCHOLTZ, M. THEOFANOS, AND B. ANTONISHEK, *Development of a test bed for evaluating human-robot performance for explosive ordnance disposal robots*, in Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction, 2006, pp. 10–17.

[156] J. SCHULMAN, P. MORITZ, S. LEVINE, M. JORDAN, AND P. ABBEEL, *High-dimensional continuous control using generalized advantage estimation*, arXiv preprint arXiv:1506.02438, (2015).

[157] J. A. SHAH, J. H. SALEH, AND J. A. HOFFMAN, *Review and synthesis of considerations in architecting heterogeneous teams of humans and robots for optimal space exploration*, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 37 (2007), pp. 779–793.

[158] H. SHEN, L. ZHAO, Y. BIAN, AND L. XIAO, *Research on SSVEP-based controlling system of multi-DoF manipulator*, in International Symposium on Neural Networks, Springer, 2009, pp. 171–177.

[159] A. K. Singh, S. Aldini, D. Leong, Y.-K. Wang, M. G. Carmichael, D. Liu, and C.-T. Lin, *Prediction Error Negativity in Physical Human-Robot Collaboration*, in 2020 8th International Winter Conference on Brain-Computer Interface (BCI), IEEE, 2020, pp. 1–6.

[160] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, *Smoothgrad: removing noise by adding noise*, arXiv preprint arXiv:1706.03825, (2017).

[161] M. Spüler and C. Niethammer, *Error-related potentials during continuous feedback: using EEG to detect errors of different type and severity*, Frontiers in human neuroscience, 9 (2015), p. 155.

[162] F. Sun, W. Zhang, J. Chen, H. Wu, C. Tan, and W. Su, *Fused fuzzy petri nets: a shared control method for brain–computer interface systems*, IEEE Transactions on Cognitive and Developmental Systems, 11 (2018), pp. 188–199.
Publisher: IEEE.

[163] M. Sundararajan, A. Taly, and Q. Yan, *Axiomatic attribution for deep networks*, in International Conference on Machine Learning, PMLR, 2017, pp. 3319–3328.

[164] E. E. Sutter, *The brain response interface: communication through visually-induced electrical brain responses*, Journal of Microcomputer Applications, 15 (1992), pp. 31–45.

[165] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, MIT press, 2018.

[166] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, *Policy gradient methods for reinforcement learning with function approximation*, in Advances in neural information processing systems, 2000, pp. 1057–1063.

[167] K. Tanaka, K. Matsunaga, and H. O. Wang, *Electroencephalogram-based control of an electric wheelchair*, IEEE transactions on robotics, 21 (2005), pp. 762–766.

[168] S. F. Taylor, E. R. Stern, and W. J. Gehring, *Neural systems for error monitoring: recent findings and theoretical perspectives*, The Neuroscientist, 13 (2007), pp. 160–172.

[169] J. Tessadori, L. Schiatti, G. Barresi, and L. S. Mattos, *Does tactile feedback enhance single-trial detection of error-related eeg potentials?*, in 2017 IEEE international conference on systems, man, and cybernetics (SMC), IEEE, 2017, pp. 1417–1422.

[170] S. Thompson, T. Horiuchi, and S. Kagami, *A probabilistic model of human motion and navigation intent for mobile robot path planning*, in 2009 4th International Conference on Autonomous Robots and Agents, IEEE, 2009, pp. 663–668.

[171] C. Tian, S. Shaik, and Y. Wang, *Deep reinforcement learning for shared control of mobile robots*, IET Cyber-Systems and Robotics, (2021).

[172] J. Tjomsland, A. Shafti, and A. A. Faisal, *Human-robot collaboration via deep reinforcement learning of real-world interactions*, arXiv preprint arXiv:1912.01715, (2019).

[173] L. Tonin, R. Leeb, M. Tavella, S. Perdikis, and J. d. R. Millán, *The role of shared-control in BCI-based telepresence*, in 2010 IEEE International Conference on Systems, Man and Cybernetics, IEEE, 2010, pp. 1462–1466.

[174] H. T. VAN SCHIE, R. B. MARS, M. G. COLES, AND H. BEKKERING, *Modulation of activity in medial frontal and motor cortices during error observation*, Nature neuroscience, 7 (2004), pp. 549–554.

[175] G. VANACKER, J. D. R. MILLÁN, E. LEW, P. W. FERREZ, F. G. MOLES, J. PHILIPS, H. VAN BRUSSEL, AND M. NUTTIN, *Context-based filtering for assisted brain-actuated wheelchair driving*, Computational intelligence and neuroscience, 2007 (2007).

[176] E. M. VENTOURAS, P. ASVESTAS, I. KARANASIOU, AND G. K. MATSOPOULOS, *Classification of error-related negativity (ERN) and positivity (Pe) potentials using kNN and support vector machines*, Computers in biology and medicine, 41 (2011), pp. 98–109.

[177] J. J. VIDAL, *Real-time detection of brain events in EEG*, Proceedings of the IEEE, 65 (1977), pp. 633–641.

[178] G. VISCONTI, B. DAL SENO, M. MATTEUCCI, AND L. MAINARDI, *Automatic recognition of error potentials in a P300-based brain-computer interface*, Citeseer, 2008.

[179] M. VÖLKER, R. T. SCHIRRMEISTER, L. D. FIEDERER, W. BURGARD, AND T. BALL, *Deep transfer learning for error decoding from non-invasive EEG*, in 2018 6th International Conference on Brain-Computer Interface (BCI), IEEE, 2018, pp. 1–6.

[180] H. WANG, T. LI, A. BEZERIANOS, H. HUANG, Y. HE, AND P. CHEN, *The control of a virtual automatic car based on multiple patterns of motor imagery BCI*, Medical & biological engineering & computing, 57 (2019), pp. 299–309.

[181] S. Wang, C.-J. Lin, C. Wu, and W. A. Chaovalitwongse, *Early detection of numerical typing errors using data mining techniques*, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 41 (2011), pp. 1199–1212.

[182] Y. Wang, X. Ye, Y. Yang, and W. Zhang, *Collision-free trajectory planning in human-robot interaction through hand movement prediction from vision*, in 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids), IEEE, 2017, pp. 305–310.

[183] L. Weaver and N. Tao, *The optimal reward baseline for gradient-based reinforcement learning*, arXiv preprint arXiv:1301.2315, (2013).

[184] R. J. Williams, *Simple statistical gradient-following algorithms for connectionist reinforcement learning*, Machine learning, 8 (1992), pp. 229–256.

[185] J. R. Wolpaw, D. J. McFarland, G. W. Neat, and C. A. Forneris, *An EEG-based brain-computer interface for cursor control*, Electroencephalography and clinical neurophysiology, 78 (1991), pp. 252–259.

[186] B. Xu, S. Peng, A. Song, R. Yang, and L. Pan, *Robot-aided upper-limb rehabilitation based on motor imagery EEG*, International Journal of Advanced Robotic Systems, 8 (2011), p. 40.

[187] D. Xu, M. Agarwal, E. Gupta, F. Fekri, and R. Sivakumar, *Accelerating Reinforcement Learning using EEG-based implicit human feedback*, Neurocomputing, 460 (2021), pp. 139–153.

[188] M. Xu, J. Han, Y. Wang, T.-P. Jung, and D. Ming, *Implementing over 100 command codes for a high-speed hybrid brain-computer interface using concurrent*

P300 and SSVEP features, IEEE Transactions on Biomedical Engineering, 67 (2020), pp. 3073–3082.

[189] M. Xu, X. Xiao, Y. Wang, H. Qi, T.-P. Jung, and D. Ming, *A brain–computer interface based on miniature-event-related potentials induced by very small lateral visual stimuli*, IEEE Transactions on Biomedical Engineering, 65 (2018), pp. 1166–1175.

[190] Y. Xu, C. Ding, X. Shu, K. Gui, Y. Bezsudnova, X. Sheng, and D. Zhang, *Shared control of a robotic arm using non-invasive brain–computer interface and computer vision guidance*, Robotics and Autonomous Systems, 115 (2019), pp. 121–129.

[191] S.-S. Yoo, T. Fairneny, N.-K. Chen, S.-E. Choo, L. P. Panych, H. Park, S.-Y. Lee, and F. A. Jolesz, *Brain–computer interface using fMRI: spatial navigation by thoughts*, Neuroreport, 15 (2004), pp. 1591–1595.

[192] T. O. Zander, L. R. Krol, N. P. Birbaumer, and K. Gramann, *Neuroadaptive technology enables implicit cursor control based on medial prefrontal cortex activity*, Proceedings of the National Academy of Sciences, 113 (2016), pp. 14898–14903.

[193] C. Zhang, Y. Kimura, H. Higashi, and T. Tanaka, *A simple platform of brain-controlled mobile robot and its implementation by SSVEP*, in The 2012 International Joint Conference on Neural Networks (IJCNN), IEEE, 2012, pp. 1–7.

[194] H. Zhang, R. Chavarriaga, Z. Khaliliardali, L. Gheorghe, I. Iturrate, and J. d R Millán, *EEG-based decoding of error-related brain activity in a real-world driving task*, Journal of neural engineering, 12 (2015), p. 066028.

[195] R. Zhang, F. Torabi, L. Guan, D. H. Ballard, and P. Stone, *Leveraging human guidance for deep reinforcement learning tasks*, arXiv preprint arXiv:1909.09906, (2019).

[196] W. Zhang, F. Sun, C. Liu, W. Su, C. Tan, and S. Liu, *A hybrid EEG-based BCI for robot grasp controlling*, in 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2017, pp. 3278–3283.

[197] X. Zhang and H. Lin, *Performance guaranteed human-robot collaboration with POMDP supervisory control*, Robotics and Computer-Integrated Manufacturing, 57 (2019), pp. 59–72.

[198] Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang, *A survey on neural network interpretability*, IEEE Transactions on Emerging Topics in Computational Intelligence, (2021).

[199] S. Zhao, Z. Li, R. Cui, Y. Kang, F. Sun, and R. Song, *Brain–machine interfacing-based teleoperation of multiple coordinated mobile robots*, IEEE Transactions on Industrial Electronics, 64 (2016), pp. 5161–5170.

[200] W. Zheng, B. Wu, and H. Lin, *Pomdp model learning for human robot collaboration*, in 2018 IEEE Conference on Decision and Control (CDC), IEEE, 2018, pp. 1156–1161.

[201] P. Zhu, X. Li, P. Poupart, and G. Miao, *On improving deep reinforcement learning for pomdps*, arXiv preprint arXiv:1704.07978, (2017).