

Article

## An Artificial Neural Network Based Robot Controller that Uses Rat's Brain Signals

Marsel Mano <sup>1,\*</sup>, Genci Capi <sup>2</sup>, Norifumi Tanaka <sup>3</sup> and Shigenori Kawahara <sup>4</sup>

<sup>1</sup> Graduate School of Science and Engineering for Education, University of Toyama, Gofuku Campus, 3190 Gofuku, Toyama 930-8555, Japan

<sup>2</sup> Faculty of Engineering, University of Toyama, Gofuku Campus, 3190 Gofuku, Toyama 930-8555, Japan; E-Mail: capi@eng.u-toyama.ac.jp

<sup>3</sup> Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology, 4259-G5-17, Yokohama-shi, Kanagawa 226-8503, Japan; E-Mail: norifumi0816@softbank.ne.jp

<sup>4</sup> Department of Life Sciences and Bioengineering, University of Toyama, Gofuku Campus, 3190 Gofuku, Toyama 930-8555, Japan; E-Mail: kawahara@eng.u-toyama.ac.jp

\* Author to whom correspondence should be addressed; E-Mail: marsel.mano@gmail.com; Tel.: +81-080-4253-8833.

Received: 27 March 2013; in revised form: 16 April 2013 / Accepted: 19 April 2013 /

Published: 29 April 2013

---

**Abstract:** Brain machine interface (BMI) has been proposed as a novel technique to control prosthetic devices aimed at restoring motor functions in paralyzed patients. In this paper, we propose a neural network based controller that maps rat's brain signals and transforms them into robot movement. First, the rat is trained to move the robot by pressing the right and left lever in order to get food. Next, we collect brain signals with four implanted electrodes, two in the motor cortex and two in the somatosensory cortex area. The collected data are used to train and evaluate different artificial neural controllers. Trained neural controllers are employed online to map brain signals and transform them into robot motion. Offline and online classification results of rat's brain signals show that the Radial Basis Function Neural Networks (RBFNN) outperforms other neural networks. In addition, online robot control results show that even with a limited number of electrodes, the robot motion generated by RBFNN matched the motion generated by the left and right lever position.

**Keywords:** brain machine interface; learning and adaptive systems; radial basis function neural controllers

---

## 1. Introduction

During the last two decades, brain machine interface (BMI) has received increasing attention. The reason being, that brain signals recorded with multiple electrodes pasted on scalp or implanted inside the brain, are regarded as a communication channel with high potential for the control of prosthetic devices [1,2]. This could be very useful to patients with disabilities, severe motor dysfunctions, locked-in syndrome, *etc.* Various BMI works, have investigated the control of robotic devices using animal or human brain activity. Trained animals have been shown to accomplish basic tasks, like getting food or water with a robotic arm by using their brain activity, recorded with brain implanted electrode arrays [3–7]. Human BMI applications include robotic arm movements, wheelchair navigation, or gaming [2,8–10]. Due to high risk and complicated surgical procedures, human invasive BMI is exploited less than animal BMI [11].

Depending on the application and the specific neural activity, the brain signals used for BMI include the Electro-Myographic (EMG), the Electro-Oculographic (EOG) and the Electro-Encephalographic (EEG). Typically, any of the above brain signals are recorded while humans (animals) do a specific mental or physical task. The signals, together with the mental activity event time logs, are used to build classifier models. These classifiers are later used to determine the mental activity online.

A common problem of all BMI applications is the acquired signal quality. The signal quality depends on the electrode location, type of signal, type of neural activity, *etc.*, but generally the recorded signal is noisy. Thus, determining specific mental activity of interest is a difficult task. Usually, in non-invasive BMI applications, this is addressed by increasing the number of electrodes and applying spatial and temporal filtering to the acquired multichannel signal [12,13]. In invasive BMI, although the signal has less noise, a high number of electrodes (generally from eight to more than 100) are required to achieve a good classification rate. In contrast to non-invasive BMI, adding electrodes through brain surgery is very complicated, risky and expensive.

In this paper, we employ a radial basis function neural network (RBFNN) in order to achieve a good classification rate with only four brain implanted electrodes. The rat's cortical activity is used to train the RBFNN, mapping the brain signal to the rat's voluntary musculoskeletal movement. In our experiment, we tested several multilayer perceptron neural network architectures and we used different training methods. The RBFNN outperformed the classification accuracy of other networks. The superiority of RBFNNs in dealing with noisy input data and selecting specific important input features makes them more tolerant to brain signals which vary from day to day and also less prone to over fitting.

For rat training, we developed a lever based robot control system. The rat's brain activity is recorded (offline) during lever based robot operation and then is used to train various artificial neural networks. After this, the network with the best classification results is used to classify the brain activity (online) and actuate the robot, while the lever functions are disabled. The RBFNN showed a higher performance and thus it was used online.

Experimental results show that the RBFNN is able to classify the rat's brain signals in real time with a high level of confidence during right and left lever press events. During two online sessions, the rat was able to get control of the robot and obtain food, only by using its brain activity. The classification accuracy achieved by the RBFNN is higher than that of similar works shown in [14].

This paper has the following structure. Section 2 explains the rat training. Section 3 presents the method used for signal acquisition, processing and the RBFNN developed for classification. The experimental setup is described in Section 4. In Section 5 the experimental results are summarized. Section 6 concludes the paper.

## 2. Rat Training

The rat used in our experiments was a five month old male Wistar/ST with electrodes pre-implanted in its brain. The surgery and all of the experimental procedures were performed in accordance with the Guidelines for the Care and Use of Laboratory Animals in the University of Toyama, and the NIH Guide for the Care and Use of Laboratory Animals.

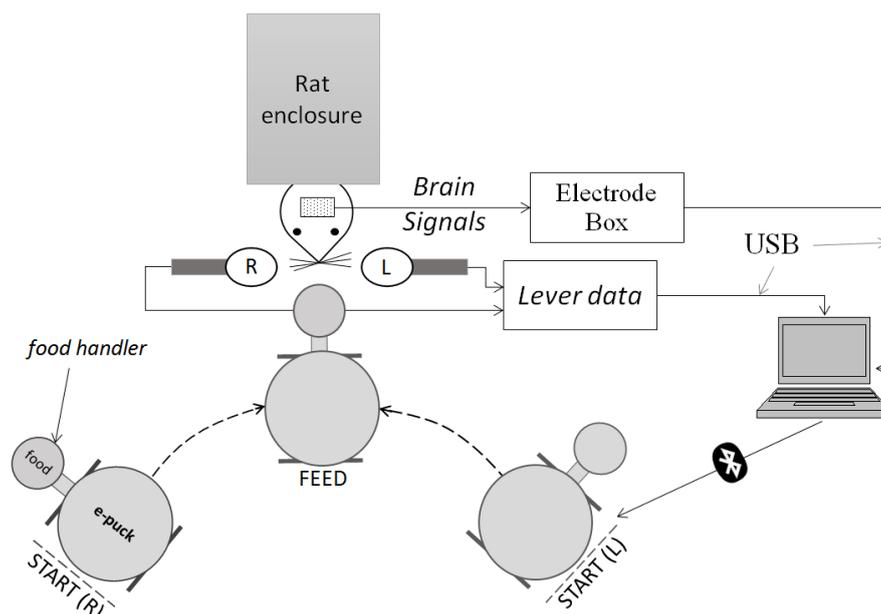
In a typical experiment, food-rewarding is used to train the rats to perform an instrumental action (*i.e.*, lever pressing) [14]. At the beginning, the rat is trained to learn to press the right or left lever as follows:

- (1) Press any lever to get food supplied manually by a human.
- (2) Press levers as above, except with their head restricted.

After the rat has learned the above tasks, the food is placed on a small robot (e-puck). The robot has a food handler (Figure 1) mounted on the left side and a green LED light under the food handler. By using the LED, the rat is able to visually detect the robot. The rat learns to direct the robot by pressing the levers in scenarios that change from simple to more complicated ones, as follows:

- (1) The robot is placed in front of the rat and it moves straight forward when any lever is pressed.
- (2) The robot is placed on the right (left) side of the rat and it follows half of a U-shape trajectory when only the right (left) lever is pressed.
- (3) The robot is placed initially on the right or left side of the rat and it moves to the right or left of a U-shape when the respective lever is pressed.

**Figure 1.** Developed system.



Training is complete when the rat learns to press the lever on the side of the LED in order to move the robot and get the food.

In our experiments, the rat was carefully placed in the rat enclosure shown on Figure 1, where it can easily access both levers. The levers are connected through a serial interface with the computer and the robot is controlled via a Bluetooth interface. Brain signals are acquired through electrodes implanted in its brain and sent to a computer.

The robot movement is in a U-shape trajectory and depending on the pressed lever it can be in either a clockwise or a counterclockwise direction. The robot initially is placed at one of the start positions (Figure 1). If the correct lever is pressed the robot moves towards the food position; if the wrong lever is pressed it moves further away. For instance, when the robot is placed on “START (R)”, the LED is on the right side of the rat. If the rat presses the right lever the robot will move clockwise to the feed position, however, if the rat presses the left lever the robot will move counterclockwise and it will go further away from the feed position. The lever control functions were enabled only during offline data recording. During online experiments the levers were used for event recording and the robot was controlled by brain signal mapped into lever actions by RBFNN.

### 3. Method

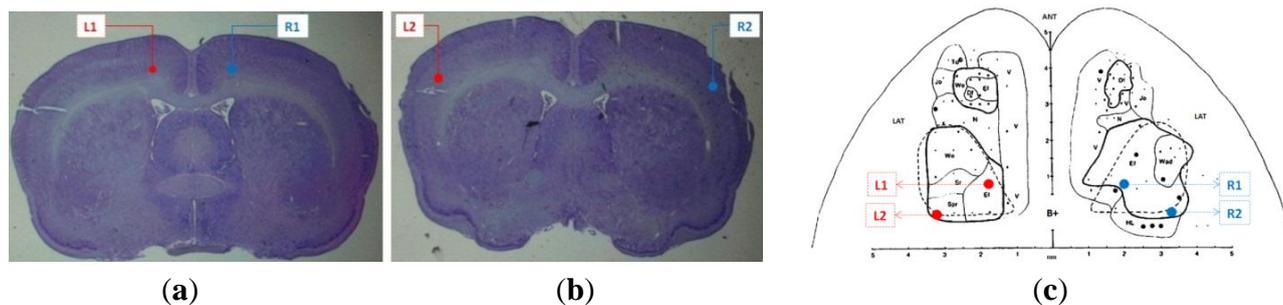
#### 3.1. Signal and Event Acquisition

Four electrodes were implanted on the rat’s brain (Figure 2).

- (1) Electrodes, L1 and R1, were implanted in the motor cortex area, respectively in the left and right hemisphere.
- (2) Electrodes, L2 and R2, were implanted in the somatosensory cortex area, respectively in the left and right hemisphere.

Electrodes were connected to the electrode box/amplifier. Next, signals were digitized at 250 Hz and collected on a personal computer via USB. All electrodes were software referenced to earth. Lever press events were simultaneously recorded.

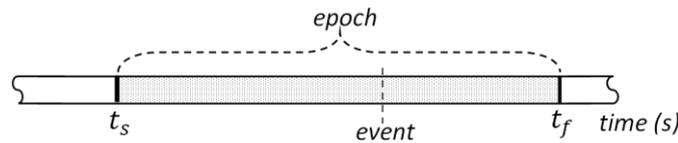
**Figure 2.** Electrode locations: (a) in motor cortex (Interaural); (b) in somatosensory area (Interaural); (c) dorsal view.



3.2. Feature Extraction and Neural Network

The offline recorded data are represented by matrix  $S \in R^{C \times T}$  where  $C$  is the number of channels ( $L1, R1, L2, R2$ ) and  $T$  the number of sample time points. Epochs  $X^{(j)}$  with duration  $E$  are extracted in the interval  $[t_s^{(j)}, t_f^{(j)}]$  in the vicinity of the lever press events, as shown in Figure 3.

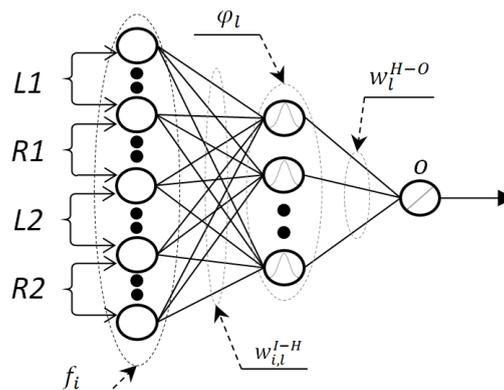
Figure 3. Epoch extraction interval.



Every epoch is then divided into  $N$  ( $j$  is omitted) time windows with duration  $(t_f - t_s)/N$ . For every window and for every channel, the sample data mean values are calculated as  $m_{c,n} = \text{mean}(X_c(t))_{t \in t_n; c \in C}$ , where  $n \in \{1, \dots, N\}$ . The data are stored in a feature vector  $F(f_i)_{(N \times C) \times 1}$ , where  $f_i = m_{c,n} \forall i = N \times c + n$ .

With all the feature vectors extracted from offline sessions, we trained feed forward neural networks with  $I = N \times C$  input layer neurons, a variable number of hidden layer neurons  $H$  and a single output neuron  $O$  (Figure 4).

Figure 4. Neural network topology.



Several multilayer feed-forward neural networks with different topologies and learning functions were tested but the RBFNN performed the best. RBFNN classification clusters are separated by hyper spheres, thus they are better local approximation networks. RBFNN outputs are determined by specific hidden units in certain local receptive fields [15]. This property, allows RBFNN to be more selective of the input features and more tolerant to error. Furthermore, during training the neurons of the hidden layer are added incrementally as the training error decreases. This provides a good initial set up for network connections with shorter training duration.

In this paper, we consider only two neural networks with similar architecture (Figure 4): an RBFNN and multi-layer perceptron neural network trained with Bayesian regulation back propagation (BPNN) [16].

During online experiments, every time a lever was pressed the brain signals were collected in the interval  $[t_s, t_f]$ . The data were similarly processed to extract the feature vector, which was classified

by the RBFNN. In online sessions, only the RBFNN was used to control the robot motion. In any case, BPNN performance was tested with the online acquired datasets, after the online sessions were finished, and the results are shown in Section 5.

### 3.2.1. RBFNN

The RBFNN's hidden layer neurons have a radial basis transfer function  $\varphi^1$ :

$$\varphi^1(v_l) = \exp\left(-\frac{\|v_l - c_h\|}{\sigma_h}\right) \quad (1)$$

And the output neuron has a linear transfer function:

$$o^1 = \sum_{l=1}^H \varphi_l^1(v_l)w_l^{H-o} + w_0^{h-o} \quad (2)$$

where:  $v_l = [f_1w_{1,l}^{l-H}, f_2w_{2,l}^{l-H} \dots f_lw_{l,l}^{l-H}]$ ,  $w^{l-H}$  and  $w^{H-o}$  are the input-to-hidden layer weights and hidden-to-output layer weights, respectively. The activation function of the hidden layer neurons  $\varphi^1$ , is a Gaussian function with center  $c_h$  and spread  $\sigma_h$ . In our RBFNN,  $\sigma_h$  is set to 1.

In a supervised learning method, neurons are added to the hidden layer of the RBFNN until it meets a specified mean squared error goal. RBFNN performance function is a mean squared error (MSE).

### 3.2.2. BPNN

In contrast with RBFNN, BPNNs have a fixed number of hidden layer neurons (30). Hidden layer neurons have a hyperbolic tangent transfer function  $\varphi^2$ :

$$\varphi^2(v_h) = \tanh(v_h) = \frac{2}{1 + e^{-2v_h}} - 1 \quad (3)$$

The output layer neuron has also a hyperbolic tangent transfer function  $o^2$ :

$$o^2 = \tanh\left(\sum_{l=1}^H \varphi_l^2(v_l)w_l^{h-o} + w_0^{h-o}\right) \quad (4)$$

Bayesian regularization is a supervised back propagation training method that updates the network weights and bias values according to the Levenberg-Marquardt optimization [17]. BPNN performance function is a summed square error (SSE).

In our implementation, the output of any neural network is classified as “left” if its value is greater than 0.5 and “right” if it is equal or less than 0.5.

## 4. Experimental Setup

In total, three offline recording sessions were done with a trained rat. All experiments were conducted in a dark room in order to make it easier for the rat to spot the LED light. The duration of the sessions varied from 20 to 30 min. The first two sessions were used to train the neural network, with random trial selection. Then the third offline session was used to test network performances and select the best network. Offline sessions were conducted on three different days over the span of a week.

During online sessions the lever information was collected for event marking purposes. However, the robot was controlled online by the RBFNN that mapped the rat’s brain signals to the lever’s position.

**5. Results**

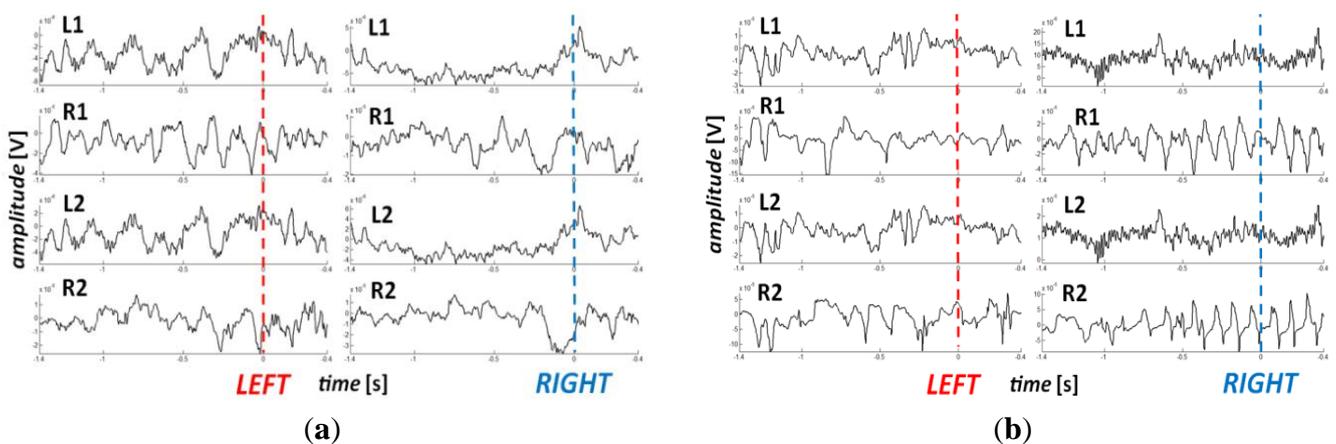
*5.1. Neural Network*

Table 1 summarizes the offline sessions that were used to train and test the neural networks. The dataset of the first two sessions contains 73 events, where 35 are left lever press events and 38 are right lever press events. For each event, epoch data were collected and processed for feature extraction. Samples of the epochs of rat’s brain signals for right and left lever press events are shown in Figure 5. The channels on the left hemisphere, L1 and L2 show similar activity to one another. The channels on the right hemisphere, R1 and R2 show less similarity but still have common spikes and dips. On the other hand, left and right hemispheres show different brain activities, but still it is very hard to linearly separate one from the other. The signals show that the brain activity started around 1.5 s before and ended around 0.5 s after the events. Different epoch duration and window lengths were tested and the segment [−1.4, 0.4] with window length 200 ms gave the best performance.

**Table 1.** Offline sessions event summary.

Session	Duration (min)	Purpose	Lever press events		
			Left	Right	Total
1	20	<i>train</i>	16	26	42
2	30	<i>train</i>	19	12	31
3	27	<i>test</i>	25	13	38
TOTAL			60	51	111

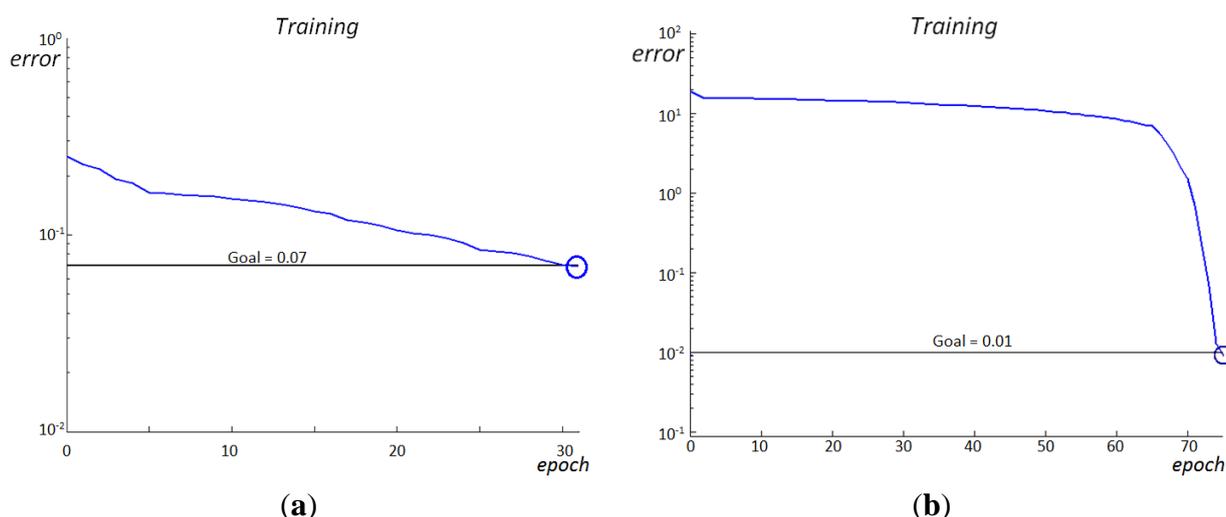
**Figure 5.** Brain signals during Left and Right lever press: (a) Single event epoch; (b) Mean of all event epochs recorded in the first offline session.



From the epoched data, input matrixes  $P_1(32 \times 73)$  and  $P_2(32 \times 38)$  together with their respective targets  $T_1(1 \times 73)$  and  $T_2(1 \times 38)$ , were created.  $P_1$  and  $T_1$  were used for supervised training of the neural networks and  $P_2$  and  $T_2$  for testing.

Figure 6 shows the training performances of the RBFNN and BPNN. In order to avoid over fitting, the RBFNN training was finished when the MSE reached the predefined goal value of 0.07. The RBFNN completed learning after 31 epochs. At the end of the training it had 31 neurons in the hidden layer. The training of the BPNN was completed after 75 epochs, when the SSE reached the predefined goal value of 0.01. The RBFNN’s selected goal value was established higher than BPNN’s since experimental results showed a decrease in performance when the RBFNN training was stressed to achieve lower goal values. Furthermore, both network goal values were determined experimentally to achieve the highest performances.

**Figure 6.** Neural network training performances: (a) Mean squared error (MSE) of Radial Basis Function Neural Networks (RBFNN); (b) Sum squared error (SSE) or Bayesian regulation back propagation (BPNN).



Both neural networks were trained and tested with the same offline sessions. Their classification accuracies are shown on Table 2. The first two sessions that were used for training, have very high classification accuracy for both neural networks. The third session, used only to test the networks, shows higher classification accuracy in the case of RBFNN compared to BPNN. Based on the results of the test session, we selected the RBFNN for the online control of the robot.

**Table 2.** Radial Basis Function Neural Networks (RBFNN) and Bayesian regulation back propagation (BPNN) classification accuracy with offline sessions.

Offline Session	RBFNN			BPNN		
	Left	Right	Total	Left	Right	Total
1 (train)	100.0	96.2	97.6	100.0	100.0	100.0
2 (train)	94.7	100.0	96.8	100.0	100.0	100.0
3 (test)	92.0	61.5	81.6	60.0	69.2	63.2

5.2. Online Robot Control

The offline trained RBFNN, was used to classify the rat’s brain signals during online robot control. The results of the online classification are summarized in Table 3.

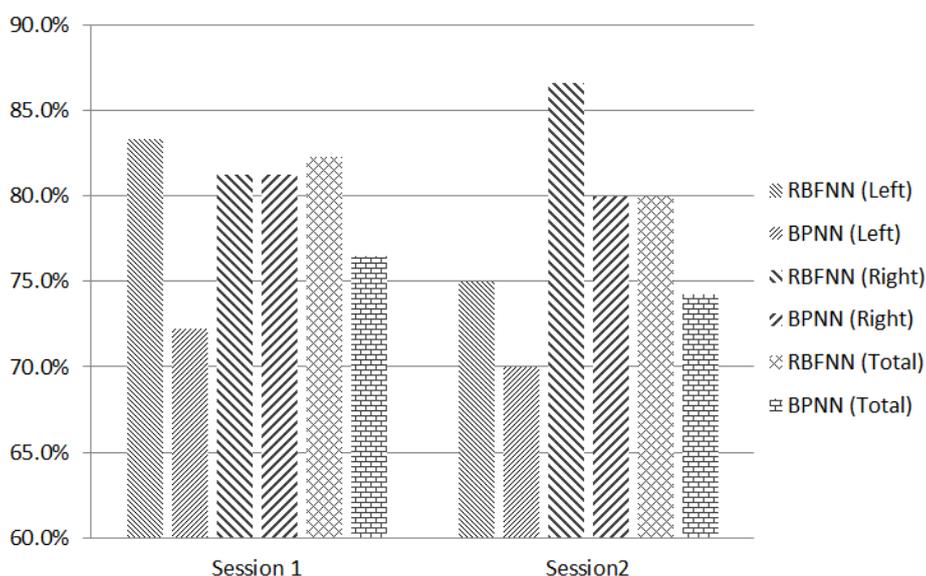
**Table 3.** Summary of online sessions.

Online Session	Duration	All Events			Misclassified events		
		Left	Right	Total	Left	Right	Total
1	36.8	18	16	34	3	3	6
2	32.7	20	15	35	4	2	6
TOTAL		38	29	69	7	5	12

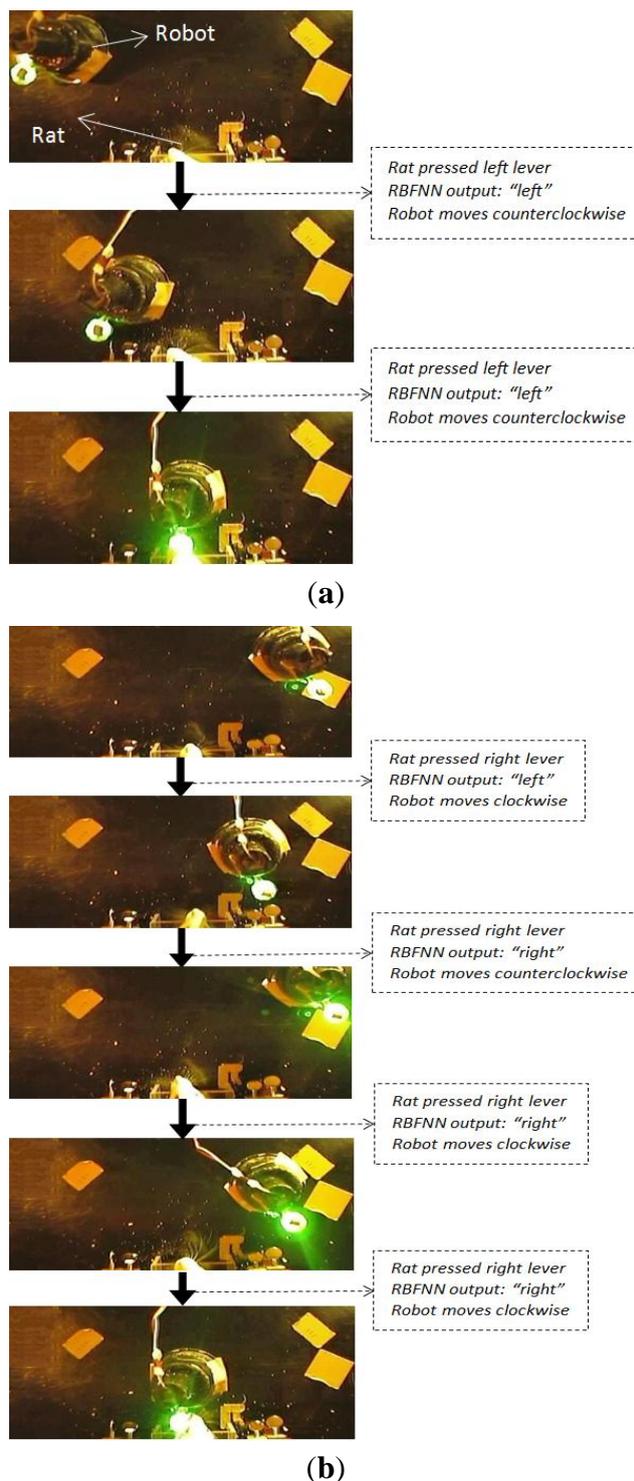
The RBFNN classification accuracy during online sessions is graphically illustrated in Figure 7. After the online sets were finished, the data acquired online was used to test the classification performance of the BPNN. As shown in Figure 7, the RBFNN performance is higher than that of the BPNN in both sessions and for each event type individually.

Figure 8(a) shows the robot motion during an experiment where the robot reached the feed position after two RBFNN outputs. The RBFNN accurately classified the brain signals and no extra steps were required. Figure 8(b) shows the robot motion during an experiment where the robot reached the feed position after four RBFNN outputs. The second RBFNN output was misclassified and the robot, instead of going to the feed position, went back to the start position. Two extra steps were needed to move the robot into the feed position.

**Figure 7.** Online session classification accuracy.



**Figure 8.** Rat’s brain controlled robot motion during: (a) a trial on the left side; (b) a trial on the right side.



## 6. Conclusions

In this paper, we presented a neural network based robot controller activated by a rat’s brain signals. In our method, we utilized the rat’s brain signals, collected by implanted electrodes, to train neural controllers. The trained neural controllers were also tested in online experiments, where the RBFNN was found to give the best performance. The results showed that even with a limited number of

implanted electrodes, the RBFNN mapped the brain signals with the robot motion (lever position) with high accuracy.

A future extension of this work would be to investigate how the electrode location influences the neural network classification accuracy. The benefits of the BMI based robot control investigation in animals presented in this paper could be used in BMI applications to restore motor function in humans.

### Acknowledgements

The authors would like to express their gratitude to the reviewers for checking this manuscript and giving their valuable suggestions.

### References

1. Wolpaw, J.R.; Birbaumer, N.; Heetderks, W.J.; McFarland, D.J.; Peckham, P.H.; Schalk, G.; Donchin, E.; Quatrano, L.A.; Robinson, C.J.; Vaughan, T.M. Brain-computer interface technology: A review of the first international meeting. *IEEE Trans. Rehabil. Eng.* **2000**, *8*, 164–173.
2. Millán, J.D.R.; Rupp, R.; Müller-Putz, G.R.; Murray-Smith, R.; Giugliemma, C.; Tangermann, M.; Vidaurre, C.; Cincotti, F.; Kübler, A.; Leeb, R.; *et al.* Combining brain-computer interfaces and assistive technologies: State-of-the-art and challenges. *Front. Neurosci.* **2010**, *4*, 1–15.
3. Taylor, D.M.; Tillery, S.I.H.; Schwartz, A.B. Direct cortical control of 3D neuroprosthetic devices. *Science* **2002**, *296*, 1829–1832.
4. Carmena, J.M.; Lebedev, M.A.; Crist, R.E.; O’Doherty, J.E.; Santucci, D.M.; Dimitrov, D.F.; Patil, P.G.; Henriquez, C.S.; Nicolelis, M.A.L. Learning to control a brain-machine interface for reaching and grasping by primates. *PLoS Biol.* **2003**, *1*, e42, doi: 10.1371/journal.pbio.0000042.
5. Velliste, M.; Perel, S.; Spalding, M.C.; Whitford, A.S.; Schwartz, A.B. Cortical control of a prosthetic arm for self-feeding. *Nature* **2008**, *453*, 1098–1101.
6. Chapin, J.K.; Moxon, K.A.; Markowitz, R.S.; Nicolelis, M.A. Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nat. Neurosci.* **1999**, *2*, 664–670.
7. Wessberg, J.; Stambaugh, C.R.; Kralik, J.D.; Beck, P.D.; Laubach, M.; Chapin, J.K.; Kim, J.; Biggs, S.J.; Srinivasan, M.A.; Nicolelis, M.A. Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature* **2000**, *408*, 361–365.
8. Yanagisawa, T.; Hirata, M.; Saitoh, Y.; Kishima, H.; Matsushita, K.; Goto, T.; Fukuma, R.; Yokoi, H.; Kamitani, Y.; Yoshimine, T. Electrographic control of a prosthetic arm in paralyzed patients. *Ann. Neurol.* **2011**, *71*, 353–361.
9. Hinterberger, T.; Veit, R.; Wilhelm, B.; Weiskopf, N.; Vatine, J.-J.; Birbaumer, N. Neuronal mechanisms underlying control of a brain-computer interface. *Eur. J. Neurosci.* **2005**, *21*, 3169–3181.
10. Ang, K.K.; Guan, C.; Chua, K.S.G.; Ang, B.T.; Kuah, C.; Wang, C.; Phua, K.S.; Chin, Z.Y.; Zhang, H. A Clinical Study of Motor Imagery-Based Brain-Computer Interface for Upper Limb Robotic Rehabilitation. In Proceedings of Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2009), Minneapolis, MN, USA, 3–6 September 2009; pp. 5981–5984.

11. Majima, K.; Kamitani, Y. An outlook on the present and future of brain-machine interface research. *Brain Nerve*. **2011**, *63*, 241–246.
12. Blankertz, B.; Tomioka, R.; Lemm, S.; Kawanabe, M.; Muller, K. Optimizing spatial filters for robust EEG single-trial analysis. *IEEE Signal Process. Mag.* **2008**, *25*, 41–56.
13. Wu, W.; Chen, Z.; Gao, S.; Brown, E.N. A hierarchical Bayesian approach for learning sparse spatio-temporal decompositions of multichannel EEG. *NeuroImage* **2011**, *56*, 1929–1945.
14. Capi, G. Real robots controlled by brain signals—A BMI approach. *Int. J. Adv. Intell.* **2010**, *2*, 25–35.
15. Xie, T.; Yu, H.; Wilamowski, B. Comparison between Traditional Neural Networks and Radial Basis Function Networks. In Proceedings of the 2011 IEEE International Symposium on Industrial Electronics (ISIE), Gdansk, Poland, 27–30 June 2011; pp. 1194–1199.
16. Haykin, S. *Neural Networks: A Comprehensive Foundation*; Griffin, J., Ed.; Prentice Hall: Upper Saddle River, NJ, US, 1999; Volume 13, pp. 409–412.
17. MacKay, D.J.C. Bayesian interpolation. *Neural Comput.* **1992**, *4*, 415–447.

© 2013 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).