Probabilistic co-adaptive brain–computer interfacing

2013 J. Neural Eng. 10 066008

(http://iopscience.iop.org/1741-2552/10/6/066008)

J. Neural Eng. **10** (2013) 066008 (15pp) doi:10.1088/1741-2560/10/6/066008

Probabilistic co-adaptive brain–computer interfacing

Matthew J Bryan[1], **Stefan A Martin**[2], **Willy Cheung**[3] **and Rajesh P N Rao**[1,2]

[1] Neural Systems Laboratory, Department of Computer Science and Engineering, University of Washington, Box 352350, Seattle, WA 98105, USA
[2] Center for Sensorimotor Neural Engineering, University of Washington, Box 37, 1414 NE 42nd St., Seattle, WA 98105, USA
[3] Graduate Training Center of Neuroscience, International Max Planck Research School, University of Tübingen, Tübingen, Germany

E-mail: mmattb@cs.washington.edu, stefan7@uw.edu, willy.cheung@student.uni-tuebingen.de and rao@cs.washington.edu

Received 16 June 2013
Accepted for publication 17 September 2013
Published 18 October 2013
Online at stacks.iop.org/JNE/10/066008

Abstract

Objective. Brain–computer interfaces (BCIs) are confronted with two fundamental challenges: (a) the uncertainty associated with decoding noisy brain signals, and (b) the need for co-adaptation between the brain and the interface so as to cooperatively achieve a common goal in a task. We seek to mitigate these challenges. *Approach*. We introduce a new approach to brain–computer interfacing based on partially observable Markov decision processes (POMDPs). POMDPs provide a principled approach to handling uncertainty and achieving co-adaptation in the following manner: (1) Bayesian inference is used to compute posterior probability distributions ('beliefs') over brain and environment state, and (2) actions are selected based on entire belief distributions in order to maximize total expected reward; by employing methods from reinforcement learning, the POMDP's reward function can be updated over time to allow for co-adaptive behaviour. *Main results*. We illustrate our approach using a simple non-invasive BCI which optimizes the speed–accuracy trade-off for individual subjects based on the signal-to-noise characteristics of their brain signals. We additionally demonstrate that the POMDP BCI can automatically detect changes in the user's control strategy and can co-adaptively switch control strategies on-the-fly to maximize expected reward. *Significance*. Our results suggest that the framework of POMDPs offers a promising approach for designing BCIs that can handle uncertainty in neural signals and co-adapt with the user on an ongoing basis. The fact that the POMDP BCI maintains a probability distribution over the user's brain state allows a much more powerful form of decision making than traditional BCI approaches, which have typically been based on the output of classifiers or regression techniques. Furthermore, the co-adaptation of the system allows the BCI to make online improvements to its behaviour, adjusting itself automatically to the user's changing circumstances.

(Some figures may appear in colour only in the online journal)

1. Introduction

Brain–computer interfaces (BCIs) (also known as brain–machine interfaces (BMIs) or neural interfaces) [1] provide a novel way for humans to interact with their environment: rather than using muscles to make movements, a person can use a BCI to translate brain activity directly into control commands for assistive devices such as a speller, prosthetic arm, or

wheelchair [2–6]. Disabled and able-bodied individuals have learned to successfully operate BCI-based devices to control a prosthetic arm [3, 5], compose messages on a computer [7], browse the internet [8], and control robotic assistive devices [9–11].

Two major challenges faced by BCIs are (1) the uncertainty inherent in decoding the user's intention from noisy and low resolution brain signals, and (2) the need to ensure that the brain and the interface adapt in a cooperative manner to achieve a common goal. The first challenge has been addressed using increasingly sophisticated signal processing and classification/decoding algorithms (see [12] for a review). However, these approaches typically commit to an action based on the single output of a classifier or decoder without taking the full uncertainty of the output into account. This could be catastrophic for a real-world BCI application such as controlling a wheelchair or prosthetic arm where an action based on an uncertain output could have dangerous consequences. In this paper, we propose an approach that chooses actions based on the full posterior probability distribution over a user's possible intentions, allowing the BCI to, for example, collect more information to reduce uncertainty before committing to an action.

The second challenge, that of co-adaptation between the brain and the BCI, has previously been addressed in two ways. The traditional approach has been to alternate between collecting data from the subject while keeping BCI parameters fixed, and adapting the parameters offline. More recent efforts have attempted to adapt the parameters of classifiers or decoders online during BCI operation with some degree of success [13–16].

In this paper, we propose a new approach to brain–computer interfacing that provides a unified framework for tackling the problems of uncertainty and co-adaptation. Our approach is based on the framework of partially observable Markov decision processes (POMDPs) [17], which to our knowledge were first used for BCI control in [18], where they were used to help with P300 identification. POMDPs allow a BCI's actions to be based on the full posterior probability distribution over the user's brain state, allowing uncertainty in brain state to be taken into account while selecting actions. Additionally, the mapping from posterior distribution to actions (the 'policy') maximizes total expected reward for a task-appropriate reward function, which we can adjust to allow the BCI to co-adapt with the user. The BCI thus becomes a cooperative partner with the user where the two try to achieve a common goal.

Through feedback from the environment, the user and BCI alter their strategies cooperatively to arrive at a solution. The BCI updates its strategy using a form of learning called 'reinforcement learning' [19], which uses feedback signals from the environment to change its strategy. We show how the BCI allows the user to search the space of possible brain states for those that work best for control in a task, while the BCI automatically infers the mapping from brain states to degrees of control.

The paper is organized as follows. We first provide an introduction to POMDPs and reinforcement-based learning (section 2). We provide a detailed description of our BCI system in section 3. In section 4, we introduce a set of simple experimental paradigms for illustrating the proposed framework using an electroencephalographic (EEG) BCI based on steady state visually evoked potentials (SSVEP) [20–22]. Our experimental results, described in section 5, demonstrate that (1) the proposed framework can optimize the performance of the BCI (in terms of speed versus accuracy) according to the uncertainty in each user's recorded brain signals, and (2) the BCI can automatically co-adapt when the user decides to change control strategies on-the-fly.

2. Introduction to POMDPs and reinforcement-based learning

The framework for BCI that we propose is based on the theory of POMDPs [17], which has its roots in the operations research and artificial intelligence communities. POMDPs provide a principled approach to decision making and action selection under uncertainty. The POMDP model assumes that the true state of an observed system (e.g., a user's brain state) is hidden but information about the state can be obtained using sensors (e.g., EEG sensors) (this is the 'partially observable' part of the POMDP). Additionally, at any point in time, the POMDP 'agent' (e.g., the BCI) can select an action (such as moving a robotic arm) which causes the hidden state to change to a new state (this pertains to the 'Markov decision process' part of the POMDP definition). The goal is to select actions so as to maximize the total expected reward. In our framework, we combine POMDPs with a reinforcement learning paradigm for learning the reward function (rather than assuming that the reward function is known *a priori* as in standard POMDP models).

2.1. POMDP definition

A POMDP is defined by a 7-tuple $(S, A, \Theta, T, O, R, \gamma)$ where:

- S is a space of possible states of the system. For the current work, we assume discrete states; but the framework can be applied to continuous states as well (e.g. [23]).
- A is a discrete set of possible actions available to the agent.
- Θ is a set of possible observations. Typically, the observations are assumed to be discrete; if the agent's sensor readings are continuous, some form of discretization is used, ranging from uniform grid-based discretization to using the POMDP model itself to inform discretization [24].
- T is the state transition model that governs the 'dynamics' of the hidden state. It is defined as a set of transition probabilities $T(s', s, a) = p(s'|s, a)$, i.e., it models the probability of transitioning from a current state s to the state s' in the next time step given that the agent takes action a.
- O is the observation model, defined as a set of observation (or 'emission') probabilities $O(o, a, s') = p(o|a, s')$. In our case, we will abbreviate this to $p(o|s')$ since our observations do not depend on the action we take.

- R is a reward function mapping a current state, action, and observation to a real-valued number: $R : (s, a, o) \rightarrow \mathbb{R}$. In the case of our particular BCI design, and for the rest of this paper, our reward function will generally be independent of the observation, i.e., $R : (s, a) \rightarrow \mathbb{R}$.
- γ is a time discount factor, which is a real number such that $0 \leqslant \gamma < 1$. The discount factor is used for mathematical reasons to allow the expected reward to converge over an infinite time horizon (see below).

Since the state is not directly observable, we keep a 'belief state' b_t defined as the posterior probability distribution over the current state given all past observations and actions. This belief is updated at each time step according to:

$$b_{t+1}(s_{t+1}) = \eta p(o|s_{t+1}) \sum_{s_t \in S} p(s_{t+1}|s_t, a) b_t(s_t) \quad (1)$$

where η is the normalization constant that ensures that the elements of b_{t+1} sum to 1. We can compute η by first calculating b_{t+1} without it. η can then be set equal to 1 over the sum of the elements of b_{t+1} before normalization. We will use this technique throughout the paper and will generally refer to constants such as η as 'normalization constants'. Note that since the equation above forms a recurrence relation (b_{t+1} depends on b_t, b_t on b_{t-1}, etc), it can be seen that each belief depends on all past observations and actions (through repeated substitution of past beliefs).

Our goal is to derive a policy $\pi : B \rightarrow A$, where B is the space of all possible belief states, which maximizes the total expected discounted reward:

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]. \quad (2)$$

Calculating the optimal policy exactly is intractable in the general case. However, a number of algorithms exist for finding approximate solutions in a reasonable amount of time (e.g., [25–27]). A summary of their performance can be found online at [28]. We use the algorithm proposed in [25] for finding the policy for our POMDP BCI since we found it to have the best performance. For implementation details of the algorithm, we refer the reader to [25].

2.2. Learning the reward function

In some circumstances the full reward function of a system is not known in advance but must be estimated from experience, an idea that has its roots in the field of reinforcement learning (see [19] for an introduction).

For the proposed POMDP BCI framework, we keep a running estimate of the reward function. This estimate is based on all actual rewards received thus far. A simple estimation method is to begin with an initial naïve reward estimate $\hat{R}$ and to incrementally update it according to:

$$\text{for } (s \in S) : \hat{R}'(s, a) = \alpha b_t(s)(r_{\text{observed}} - \hat{R}(s, a)) + \hat{R}(s, a) \quad (3)$$

where a is the action that was executed, r_{observed} is the actual reward received, and α is a 'learning rate' ($0 < \alpha \leqslant 1$) which controls the sensitivity of our estimate to new information.

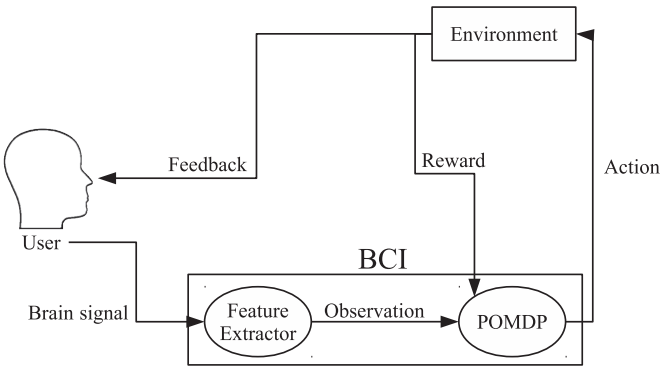


Figure 1. POMDP BCI for probabilistic co-adaptive control. The POMDP model selects actions based on the entire posterior probability distribution over brain states so as to maximize total expected reward. The 'Environment' here denotes the application being controlled, such as a robot, wheelchair, cursor, or menu system.

Since we know the action but not the state the system was actually in, we weight the magnitude of our updates according to the belief $b_t(s)$ for each state s. We will extend this logic further for our BCI system below.

3. BCI system description

3.1. Overview

In the BCI described in this paper (see figure 1), the POMDP is used to explicitly deal with uncertainty associated with the user's brain state and to decide the amount of information that must be collected before being confident enough to make a control decision. We also use reinforcement-based learning (see above) to allow our system to co-adapt with the user. The user decides on the goal of the system (e.g., the direction to steer a wheelchair), and the control mapping (e.g., SSVEP channel 1 maps to steering left). The BCI learns this control mapping automatically through reinforcement and actuates the control of the system. The user can then monitor the progress of the system and make further adjustments as necessary, to which the BCI in turn adapts based on reinforcements received. Because of this co-adaptive behaviour, the user is able to search through the space of possible brain states and to use them to form the control mapping that they are most satisfied with. As contexts change, the user may change their control mapping and the BCI system will adapt accordingly.

For our proposed framework to be applicable, a BCI control problem must satisfy a few properties.

(i) The state, action, and observation spaces must be discrete (or if continuous, they need to be discretized in some manner).
(ii) The connection between the hidden state of the system and the observations must be quantifiable.
(iii) The problem must allow control in discrete time steps (or mechanisms such as online discretization and interpolation should be employed).
(iv) For co-adaptation, the user and the BCI must have joint awareness of feedback from the environment, allowing

them both to evaluate outcomes against an accepted set of desirable states. This feedback may be given even before the erroneous action is performed. For instance, if a user steers a wheelchair in a hallway, and the BCI attempts to turn it into a wall, rather than actually causing a collision, the system could simply produce a negative feedback signal (or penalty) to allow the BCI to correct itself. Such an approach assumes an environment model and proximity sensors. Alternatively, one could obtain a negative feedback signal from the user's perception of imminent errors. It has been shown (e.g., [13]) that error perception can be robustly detected in a user and employed to improve BCI performance.

3.2. SSVEP BCI

Our BCI is based on EEG, a popular non-invasive recording technique that involves placing electrodes on the scalp to measure electrical potentials generated by synaptic inputs to large populations of neurons lying under the electrodes (see [1] for a review of EEG-based BCIs). The BCI paradigm we use is based on SSVEPs [21], which are oscillatory potentials observed in EEG recorded from the occipital (visual cortical) areas of the brain when the subject is focusing on a flickering visual stimulus (e.g., LED flashing at 15 Hz).

3.2.1. BCI hardware. Our SSVEP-based BCI supports up to five separate frequencies. We use LEDs as stimuli due to the freedom in frequency selection (rather than an LCD monitor as in a previous set-up [29]). We mounted red LEDs (within circular light boxes) around the lower perimeter of an LCD monitor. The light boxes were mounted behind a diffusive material to enhance the stimulus effect. We used five frequencies: 12, 15, 17, 20, and 22 Hz, though the number of frequencies we actually used for a given experiment varied by experiment.

We recorded the EEG signal using a g.tec USBamp EEG recording system (Guger Technologies, Austria). Gold-plated electrodes were placed at two standard locations under the 10–20 convention: O_z, located over the visual cortex (centre of back of the head) and F_{pz} (frontal location somewhat above where the eyebrows meet). We use F_{pz} as ground, and to remove artefacts caused by blinking. The sampling frequency was 256 Hz.

3.2.2. Feature extraction. We apply a fast-Fourier transform (FFT) to 1.0 s time windows, which have 0.5 s overlap. We keep only the frequencies at which our stimuli of interest are flashing—this becomes our feature vector. We experimented with other feature vectors, such as augmenting the vector with harmonics of the LED frequencies, but found that these did not improve performance in any discernible way.

Once obtained, we normalize the feature vectors by the subject's baseline level of EEG activity. Before starting the main portion of each experiment, we have the user fixate on a cross in the centre of the screen for 10 s, with the stimuli on and visible in their peripheral vision. We once again calculate 1.0 s FFT windows, and average them over the 10 s. We use this baseline data to normalize our training data before discretization. Normalization in this case is the element-wise division of the training datum and the normalization vector.

3.2.3. Observation model and discretization method. The features we obtain from EEG reside in continuous space. However, the traditional POMDP model requires a discrete observation space. As a result, we need to utilize a discretization method. The easiest and most commonly used approach is uniform discretization, which means uniformly dividing each dimension of the space within some finite bounds. This results in a grid-like partitioning of the continuous space. A major problem with uniform discretization is that one has to choose a fixed size for all grid cells: if the size is too big, resolution is lost in the feature space and the POMDP is unable to take distinct actions for different features. On the other hand, too fine a resolution results in a large number of cells, causing the POMDP model to scale poorly to large feature spaces.

In some cases, a more informed method of discretization can be employed. For example, the POMDP's planning process itself can be used to inform discretization [24]. However, we expect this method to be too slow for an online POMDP-based system such as ours. As a result, we propose a discretization method that uses *a priori* information about the observation function to achieve a non-uniform discretization of the feature space. We break the process up into three parts.

(i) Build an approximation to the observation function using a set of Gaussian distributions—one for each state in S.
(ii) Use this Gaussian model to make a non-uniform partitioning of the feature space. Each partition is a bucket into which a continuous observation may fall, and the discrete label of that observation will be the partition's label.
(iii) Numerically integrate each of the Gaussians within each partition to derive the discretized observation function $p(O|S)$.

We begin by estimating a Gaussian distribution for our features given each brain state. Recall that our feature space has one dimension for each SSVEP channel and represents the power of the EEG signal at each of those frequencies. Within that space we fit a Gaussian for each state based on the user's training data, which gives us a continuous approximation to $p(O|S)$.

Next, we form partitions in the space where all points yield roughly equivalent 'evidence', which is a quantity we define in terms of equation (1). Since we do not make any observations once we reach a final state, and since the other transitions leading to non-terminal states are deterministic, equation (1) reduces to:

$$b_{t+1}(s_{t+1}) = \eta p(o|s_{t+1}) b_t(s_{t+1}). \tag{4}$$

We define 'evidence' as:

$$\text{evidence} = \frac{b_{t+1}(s_{t+1})}{b_t(s_{t+1})} = \eta p(o|s_{t+1}). \tag{5}$$

Note that under this definition, observations o that yield equivalent evidence have vectors $p(o|S)$ which are scalar

multiples of each other. An equivalent way of stating this is that for a given prior belief b_t, two observations yield equivalent evidence if they yield the same posterior belief b_{t+1}. For example, suppose we have two states and our prior belief is:

$$b_t = \begin{pmatrix} p(s_1) \\ p(s_2) \end{pmatrix} = \begin{pmatrix} 0.3 \\ 0.7 \end{pmatrix}. \quad (6)$$

Suppose that two possible observations give

$$\begin{pmatrix} p(O|S = s_1) \\ p(O|S = s_2) \end{pmatrix} = \begin{pmatrix} 0.0025 \\ 0.0075 \end{pmatrix} \quad (7)$$

and

$$\begin{pmatrix} p(O|S = s_1) \\ p(O|S = s_2) \end{pmatrix} = \begin{pmatrix} 0.005 \\ 0.015 \end{pmatrix}. \quad (8)$$

In both cases, our posterior belief is

$$b_{t+1} = \begin{pmatrix} 0.125 \\ 0.875 \end{pmatrix}. \quad (9)$$

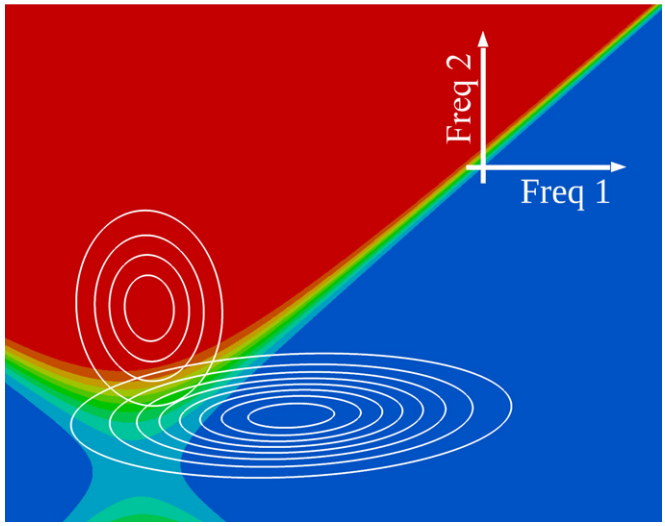
We would say that these two observations yield equivalent evidence.

To calculate our partition based on evidence, we first create a high-resolution grid in our feature space, within some finite bounds. This grid is chosen to be of far higher resolution than we would reasonably use in the context of a POMDP. Its bounds are a rectangle with sides a large Mahalanobis distance [30] from the means of the Gaussians for all classes. We found that we could achieve a relatively high-precision discretized estimate of the observation function given wide enough bounds (e.g., a Mahalanobis distance of 3) and a reasonable spatial resolution (e.g., 1 million grid points).

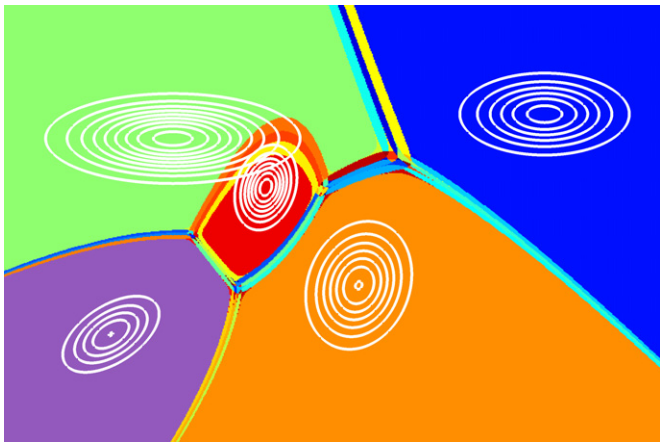
Next, we calculate the vectors $p(o|S)$ at each of these grid points o, which we normalize to sum to 1. As discussed above, equivalent points have $p(o|S)$ vectors which are scalar multiples. We choose a common scale, which we set for convenience to 1. We call this normalized vector an 'evidence vector'. We give each element of the evidence vector an integer label in $[1, n]$, where a higher n corresponds to a higher resolution discretization. To label the ith element of the evidence vector, which we denote as evidence$_i$, we use: label$_i$ = round$((n - 1) *$ evidence$_i + 1)$. The output of this process is a vector of labels for each point in the grid. Finally, we assign a discrete observation label to every unique value these vectors take. That is—every grid point o with the same vector value is considered to be in the same discrete region. This gives us the desired non-uniform discretization of the observation space. Figure 2(a) illustrates the non-uniform discretization obtained using this method for a two-dimensional observation (i.e., feature) space.

Our method generates more discretized areas as the dimensionality of the observation model increases. To prevent combinatorial explosion, we employ a k-means clustering method among the discretized areas based upon their centroids. This gives us precise control over the dimensionality of the final observation model.

To compute the observation probabilities $p(o|S)$ after discretization, we perform numerical integration within each of the regions for each class' Gaussian probability density function (PDF), using the same bounds and spatial resolution as our original grid.



(a)



(b)

Figure 2. Subject-specific discretization of a continuous feature space for a discrete POMDP observation model. (a) Discretization of a two-dimensional feature space (here, frequency 1 and frequency 2 in the SSVEP paradigm) for user 1 for a two-class state space. Level curves of the Gaussian probability density function (PDF) for each class appear in white. We used a discretization size of ten cells. Colouring indicates the discretized areas for this user. (b) Example discretization with five arbitrary classes in a 2-D feature space.

The proposed method also generalizes to a larger number of classes, as illustrated in figure 2(b) which shows the discretization for five arbitrarily formed classes in a two-dimensional feature space.

Our method has two beneficial effects. First, any two points within the same discrete area yield evidentially similar information. If we were to split this area in a way that puts the points in separate regions, that would add unnecessary model complexity. On the other hand, if we were to combine points from different regions, they would appear identical to the POMDP despite providing different information, thus potentially affecting performance. Second, as we see in figure 2(a), the spatial resolution of the discretization is higher in regions of higher uncertainty. This gives us finer distinctions where we need it, and fewer distinctions where we do not need it, suggesting effective use of the model complexity. As a result, this informed method of discretization allows

us to minimize aliasing errors while also minimizing model complexity.

3.3. POMDP model

For the experiments, the POMDP BCI was designed to track the current level of uncertainty regarding the user's brain state (e.g., whether the desired choice is SSVEP frequency 1 or frequency 2). Based on the reward function, the BCI automatically learns the threshold at which it has collected enough information to be confident about the user's brain state.

The amount of information required to make the selection depends on the relative importance of making a correct decision and the cost of waiting to collect more information. The POMDP model thus provides a principled way to balance accuracy and speed. This balance is typically user- and application-dependent, and these preferences are specified through the POMDP's reward function.

The speed with which the system makes a decision also depends on the actual observations made and the expected usefulness of future observations. This is because the entropy of the distribution governing the observation function determines how useful observations will be in making the control decision. Observations from a high entropy model provide less useful information, so the agent is forced to collect more of these on average in order to make a decision, or accept making a less confident decision.

3.4. POMDP for SSVEP BCI

3.4.1. POMDP specification. We define the components of the POMDP for the SSVEP BCI as follows.

- The POMDP's state space consists of the user's choices, each associated with a specific flashing SSVEP stimulus. We also utilize 'terminal states', one for each of the degrees of freedom of control. A terminal state is entered when the corresponding control action is selected. The associated action is then executed, the trial is ended, and the system resets for the next trial. Thus, to summarize, the POMDP has one state for each available SSVEP stimulus, and one terminal state for each possible control action. For the first set of experiments described here, the two control actions, associated with the two terminal states, are 'left' and 'right' (assumed to represent two options in a menu or directions of motion for a semi-autonomous wheelchair). The two other states represent the SSVEP stimuli associated with the 15 Hz and 20 Hz LEDs. Thus, the POMDP's state space can be defined as: $S = \{s_{15\,\mathrm{Hz}}, s_{20\,\mathrm{Hz}}, s_{\mathrm{left}}, s_{\mathrm{right}}\}$, where the latter two states are terminal states. As described below, we also performed other experiments with varying numbers of stimuli and possible controls.
- In the simplest experiment, the POMDP's action space includes the two control actions ('left' and 'right') as well as a 'wait' action: $A = \{a_{\mathrm{wait}}, a_{\mathrm{left}}, a_{\mathrm{right}}\}$. Selecting the 'wait' action allows the BCI to collect an additional time window of EEG data, giving it more information with which to base its control decision. Other experiments utilized additional possible controls.

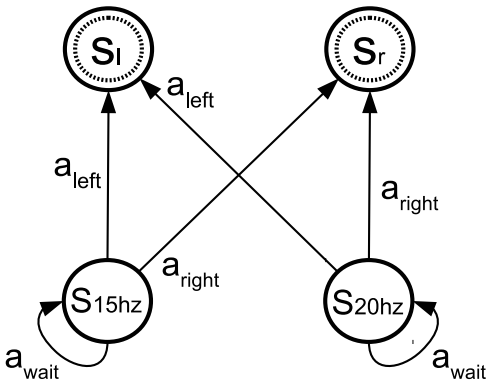


Figure 3. State transition diagram for the POMDP BCI used in the first set of experiments. The user fixates on either the 15 Hz or 20 Hz stimulus, represented as $s_{15\,\mathrm{Hz}}$, $s_{20\,\mathrm{Hz}}$ respectively. The BCI waits as data arrives (action represented by a_{wait}). Once it feels confident enough to make a control decision (a_{left}, a_{right}), the system transitions to a terminal state s_L or s_R. At this point the system resets.

- The POMDP's observation space is obtained by discretizing the EEG feature vector space as described above in section 3.2.3. After discretization, the observation space corresponds to discrete regions of the EEG feature space labelled generically through enumeration: $\Theta = \{o_1, o_2, \ldots o_n\}$.
- The transition model $T(s', s, a) = p(s'|s, a)$ in the present implementation is deterministic, though the general POMDP model supports stochastic transitions between states. Rather than enumerating the transitions, we refer the reader to figure 3, which depicts the transition model graphically. In future work, we intend to utilize more sophisticated transition models for hierarchical task modelling (see, e.g., [29, 31, 32]).
- The POMDP's observation model depends highly on the specific user, as does the discretization of the user's feature space. Since the cardinality of the observation space varies with the user, we will not depict a specific observation model here. As noted above, the observation model is a conditional probability matrix $O(o, a, s') = p(o|a, s')$.
- The reward function R is a matrix $R : S \times A \rightarrow \mathbb{R}$. The BCI automatically adapts the reward function during our co-adaptive experiments as it amasses experience through reinforcement learning. This reinforcement allows the system to learn the user's intended mapping from visual stimuli to control. In the other experiments, we fix the reward function (and therefore a control mapping) in order to evaluate the POMDP model's behaviour. In these cases, we use a set of generic parameters: r_{success}, r_{failure}, r_{wait}. For example, we may choose the values $r_{\mathrm{success}} = 10$, $r_{\mathrm{failure}} = -50$, $r_{\mathrm{wait}} = -2$. This particular example emphasizes the importance of avoiding inadvertent control by penalizing incorrect control far more than it rewards correct control. These generic parameters lead to the fixed R shown in table 1.
- We define the time discount factor to be: $\gamma = 0.99$.

Table 1. Reward function (for all users).

	a_{wait}	a_{left}	a_{right}
$S_{15\,\text{Hz}}$	r_{wait}	r_{failure}	r_{success}
$S_{20\,\text{Hz}}$	r_{wait}	r_{success}	r_{failure}
S_{left}	0	0	0
S_{right}	0	0	0

3.5. Reinforcement-based learning and co-adaptation

We turn our basic POMDP-based control model into a co-adaptive control model by adding a feedback mechanism. We attempt to infer the user's intended control mapping (here, which stimulus should map to which control output) using feedback from the environment. This feedback tells us whether the cooperation between the user and the BCI was successful at accomplishing a task.

The result of this reinforcement process is that the BCI will identify the user's intended control mapping over time. If the two do not agree on a control mapping, the BCI will receive adverse feedback from the environment, which will encourage it to try a different control mapping. Eventually, they will synchronize and control will be as successful as the Bayesian inference and state estimation allow. Also, since reinforcement will still continue to be received, the user has the freedom to switch control mappings at any time in the future. If they switch to performing a different task and feel a different control mapping would be in order, they could simply begin to use it and the BCI would eventually adapt and identify it.

We will demonstrate this on some generic tasks in section 5, but for now, let us consider a simple example. Suppose we want to design a BCI for a wheelchair control. The user would prefer that the wheelchair does not collide with objects, and therefore, the BCI has that preference as well. Finally, suppose the wheelchair is equipped with sensors to detect imminent collisions. In such a system, if a collision is imminent, the wheelchair stops automatically, and both the user and the BCI receive feedback from the wheelchair that an undesirable outcome could have resulted from the BCI's control. The co-adaptive system can then use this feedback to update its control model.

To implement our system, we begin with three straightforward assumptions. First, we assume that all available brain states are mapped to some control output, i.e., if a user is in some brain state and the BCI manages to identify it, then some control output will be actuated. Second, we assume that all possible control outputs map to at least one brain state, i.e., we will never choose a control mapping that makes some control outputs unavailable to the user. Finally, we assume that the user or system designer can express the relative cost of failure and the benefit of success. We express these as real numbers r_{failure} and r_{success}.

We demonstrate three co-adaptive systems which use these three assumptions. In the first, we have the simplest of our systems—we map two SSVEP frequencies to two possible control outputs. In our second experiment, we show how this scales to five frequencies and two outputs. This change allows the user to search the space of possible input methods for the ones that suit them the best. Finally, in a third experiment we use three frequencies and three outputs. This experiment differs from the others in its use of a reinforcement learning technique known as 'exploration'.

3.5.1. The simple case—two frequencies and two outputs. Many reinforcement systems require 'exploration' where the system tries out different actions to gather information. Since in an unconstrained system, reinforcement only applies to the action taken in a given state, it may need to try several different actions in order to estimate the various elements of the reward function. This can be contrasted with 'exploitation' where, given the current estimate of the reward function, the agent simply performs the action that maximizes the total expected future reward.

Given our three assumptions enumerated above, we can show that in many cases, including the simple case described here, exploration is not necessary. This is because our assumptions impose constraints on the values of the reward function. First, note that there are only two admissible control mappings given our assumptions: one maps the 15 Hz channel to the 'left' command, and the 20 Hz channel to the 'right' command. The other mapping reverses this. Suppose that the BCI identifies the current state as $s_{15\,\text{Hz}}$, chooses action a_{left}, and receives feedback r_{success} from the environment. Since control was successful in this case, we can conclude that exploiting the corresponding mapping from $s_{20\,\text{Hz}}$ to a_{right} would also be a success. Furthermore, since we know the cost of failure r_{failure}, we can also update the two elements of the reward function referring to the other control mapping. As a result, regardless of which control decision we make and which feedback we get, we will effectively receive the same information and will be able to update all four of these elements of the reward function.

The argument above implies that we are free to take the reward-maximizing action. Since there is no information that we can gather through exploration which we could not get through exploitation, exploitation is the strictly better choice. This does not imply that we no longer need to collect information; instead it states that what information we collect through exploitation will always be at least as good. We will always need to collect information because of the uncertainty in the reinforcement process, and because of the potential non-stationarity in the user's control strategy.

Our reinforcement algorithm for updating the reward function is built around this understanding. When the BCI receives a reinforcement signal after an action, it updates all four elements of the estimated reward function. For every element of the reward function corresponding to the successful control mapping, we update the estimated reward function using:

$$\hat{R}'(s,a) = \alpha\beta(r_{\text{success}} - \hat{R}(s,a)) + \hat{R}(s,a) \tag{10}$$

where (s,a) is a member of the successful control mapping (e.g., $(s_{15\,\text{Hz}}, a_{\text{left}})$ and $(s_{20\,\text{Hz}}, a_{\text{right}})$), α is the learning rate, and β is the confidence we have in our classification of the current state s; here, we assume that the current state is the most likely state according to our current belief state b_t and therefore, $\beta = \max(b_t)$. Once we have made this update, we

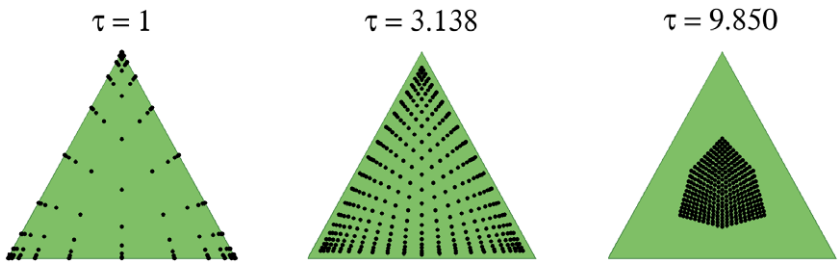


Figure 4. Effect of Boltzmann temperature (τ) on propensity to explore. As τ increases, the grid of points moves closer to the centre to indicate more uniformity in the distributions. This indicates a greater propensity to explore since differences in expected reward between the actions have less impact on the action selection probabilities.

repeat the same process but for all (s, a) pairs corresponding to the other possible control mapping, after substituting r_{failure} for r_{success}. Note that we use this max-based method rather than the one in equation (3) to take advantage of the control mapping constraints in the simple case being discussed in this section.

3.5.2. Providing more choices—five frequencies and two outputs. We repeat the experiment above but with five frequencies instead of 2. This gives the user an opportunity to search through the space of possible inputs for those that work best. Unlike the previous experiment, in this case, we will not be able to update every element of the reward function in each trial. For instance, if we receive negative feedback on a given trial, that does not tell us which of the other states should map to that output.

As with the first experiment, we do not make explicit use of exploration, but the user is implicitly engaged in a form of exploration. Suppose that the user chooses the 15 Hz channel with the intent of actuating an a_{left} control. If positive feedback is received, we can give positive reinforcement to the $(15\,\text{Hz}, a_{\text{left}})$ mapping, and a negative reinforcement to $(15\,\text{Hz}, a_{\text{right}})$ using the same mathematical logic as above. As the user explores the various input channels, each will independently learn a mapping to a control output.

3.5.3. Exploration, and generalization to an under-constrained system—three frequencies and three outputs. This final experiment shows how we can generalize the reinforcement system to a case where our assumptions are not as highly constraining. In this case, exploitation does not always yield the necessary information, so exploration by the BCI is useful. Additionally, the modifications we can make to the reward function depend on whether the reinforcement we receive is positive or negative. This means our reinforcement logic is slightly different.

First note that our assumptions result in a one-to-one control mapping. If we receive a negative reinforcement, we can make a downward adjustment to the (s, a) pair which we attempted, but we cannot make any adjustments to any other pairs since there are still multiple remaining actions to which that state could be mapped, and multiple remaining states to which that action could be mapped. On the other hand, if we receive a positive reinforcement, we can not only make an upward adjustment to that (s, a) pair, but we can make a downward adjustment to all other pairs involving that same state or action. This is precisely how we make adjustments with this system, again using the same mathematical logic as above.

In order to explore, we modify the process of selecting an action. Rather than strictly taking the action that appears most rewarding, we instead choose an action from a probability distribution, where the most rewarding is the most likely to be chosen. The POMDP continues to decide when enough evidence has accumulated to take an action, and this stochastic process is then used to choose which action that will be.

We use a commonly used exploration strategy in reinforcement learning, namely, using a Boltzmann distribution over actions for selecting an action. This strategy is sometimes referred to as 'softmax action selection' [19]. First, we compute a weighted sum of the rows of the reward function, where the weights are the elements of the current belief state (refer to table 1). This gives us a vector A_w of expected rewards for each action:

$$\text{for } (a_i \in A) : A_w(a_i) = \sum_{s_i \in S} b(s) R(s, a_i). \tag{11}$$

These action values are used to derive the Boltzmann distribution over actions:

$$P(a_i) = \frac{\exp \frac{A_w(a_i)}{\tau}}{\sum_{a_j \in A} \exp \frac{A_w(a_j)}{\tau}} \tag{12}$$

where τ is known as the 'temperature' parameter and represents the BCI's propensity to explore rather than exploit.

Higher values of τ result in a larger degree of exploration on average. To illustrate this property, one can generate a uniform grid of inputs to the Boltzmann function and see how the resulting probability distributions vary as a function of τ. In figure 4, we display these points in a simplex, which represents the space of all multinomial distributions over three categories. A point in the simplex represents a single distribution. Specifically, a point in the centre represents the uniform distribution. Points in the corners represent certainty of one of the three possibilities. As shown in figure 4, when τ increases, the points move closer to the centre, indicating more uniformity in the distribution. In our context, this means the various actions are close to being equally likely to be chosen, and differences in expected rewards between them are less significant.

4. Experimental setup

To test these models, we recruited ten subjects (eight male, two female). Five of them were engineering students, and all of them were between the ages of 20 and 30. Only one subject had previous experience with a BCI. Our experiments were approved by the University of Washington Institutional Review Board, and all subjects gave informed consent prior to the experiments. Subjects were paid US$10/hour for their participation.

In the first session, each subject performed a simple open-loop experiment to collect training data (figure 5(a)). The session consisted of six trials per state (12 s per trial). An arrow was used as a cue to instruct the user to focus on one of two SSVEP stimuli, picked at random in each trial. The training data collected consisted of power features as described above, calculated over 1 s windows with a 0.5 s overlap. This training data was used to train the POMDP observation model. In the second session, we recorded the baseline reading of each user's EEG signals.

We performed six additional sessions, during each of which we recorded six trials for each state/stimulus. This data was used as the test data set for our offline experiments. Offline analysis allowed us to test the effects of varying the POMDP's parameters as well as to evaluate a baseline Bayesian Gaussian classifier for comparison with the POMDP model. Specifically, in the offline analysis, we first tested a Bayesian Gaussian classifier, which was based on a Gaussian estimated from a fixed-size 5 s time window of data from each trial. We also repeated this for 3 s time windows in a separate experiment. We trained the observation model for a POMDP and ran it on the same test data but allowed it to automatically choose the size of the data window (by successively selecting the 'wait' action until a decision is made). We chose the POMDP's parameters (reward function, time discount factor) to obtain a balance between speed and accuracy. We used the same parameters for all subjects, though in practice, we would want to choose these parameters to fit the individual user's preferences.

In a subsequent session (figure 5(b)), we invited five subjects to return for a set of closed-loop co-adaptive experiments. We picked subjects with a wide range of classification accuracies. As before, we recorded a training data set and a baseline for normalization. Once this was complete, we asked subjects to pick a specific control mapping of their choice for the experiment. The experiment consisted of 12 trials—six with a 'left' target and six with a 'right'—in a randomized order. At the end of this experiment, we asked them to switch their control mapping and then repeated the experiment. This allowed us to test the ability of the co-adaptive system to automatically discover the user's initial control mapping, and then to detect the user-induced change in the control mapping. We performed this set of experiments twice for each user.

In the second set of co-adaptive experiments, we invited three users to perform a more ambitious version of the co-adaptive paradigm involving five SSVEP channels instead of 2. We investigated whether the POMDP BCI could co-adapt with the user to simultaneously converge to the appropriate mapping desired by the user.

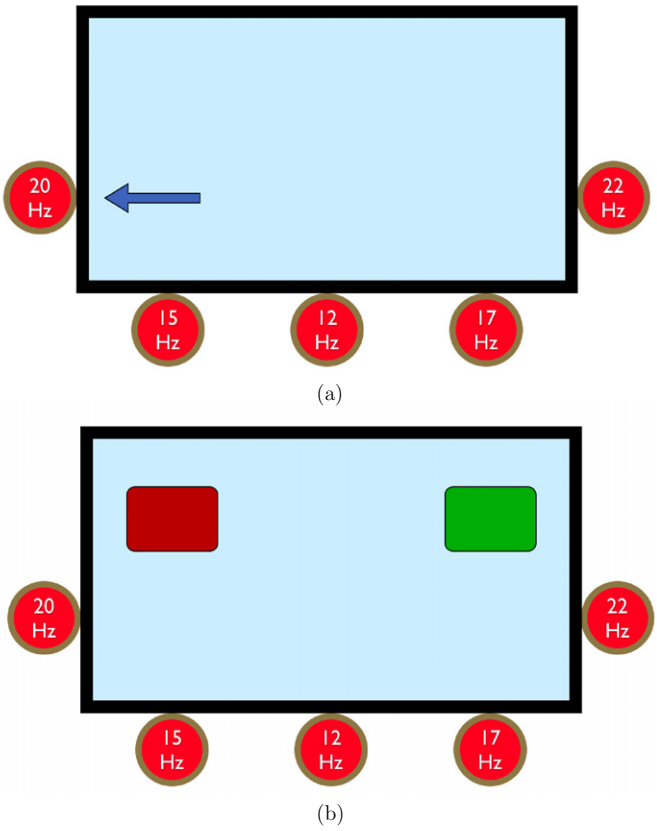


Figure 5. POMDP BCI experiments. Red LED stimuli are arrayed on the perimeter of the screen, with trial cues ('left' and 'right') displayed on the screen. For our 5-channel experiments, all five of these SSVEP channels were used. For the 2-channel experiments, only the 15 Hz and 20 Hz were used, but all five were on. Likewise, for the 3-channel experiments, 12 Hz, 15 Hz, and 17 Hz were used. (a) In the first session (data collection phase), the user was given cues indicating which stimulus to focus on. (b) In the co-adaptive experiment, the user performed a selection task; their goal was to select the green box from among the various boxes displayed ('right' in this case) by consistently focusing on one of the LED stimuli without disclosing their choice *a priori*. In the case of three boxes, an additional box appeared in the lower centre.

Finally, we invited three users to perform the final set of co-adaptive experiments. This set involved three input channels and three targets. We asked the user to choose an initial mapping, and twice during the experiment, we asked them to change the mapping. This allowed us to demonstrate the use of exploration by the BCI for co-adapting with the users' non-stationary control mappings.

5. Results

5.1. Speed–accuracy trade-off in the POMDP BCI

Our offline experiments demonstrated that the POMDP BCI's decision time for the SSVEP task varies from one user to another according to the uncertainty in their EEG signals (table 2). This variation in response time arises because the BCI withholds its control decision until the confidence threshold for the user (as dictated by the optimal policy) is reached.

Table 2. Accuracies and average decision times by user and decision algorithm.

	Gaussian fixed-size window (5 s)	Gaussian fixed-size window (3 s)	POMDP	
User ID	Model accuracy (%trials correct)	Model accuracy (%trials correct)	Model accuracy (%trials correct)	Model average decision time (s)
1	100.0	98.6	94.4	1.430
2	95.8	88.9	90.2	2.090
3	69.4	66.7	80.5	5.340
4	88.8	86.1	90.2	3.888
5	87.5	80.6	80.5	1.930
6	81.9	77.8	84.7	5.416
7	86.1	83.3	84.7	3.000
8	77.7	69.4	81.9	5.715
9	56.9	55.6	59.7	4.812
10	84.7	83.3	90.2	4.534

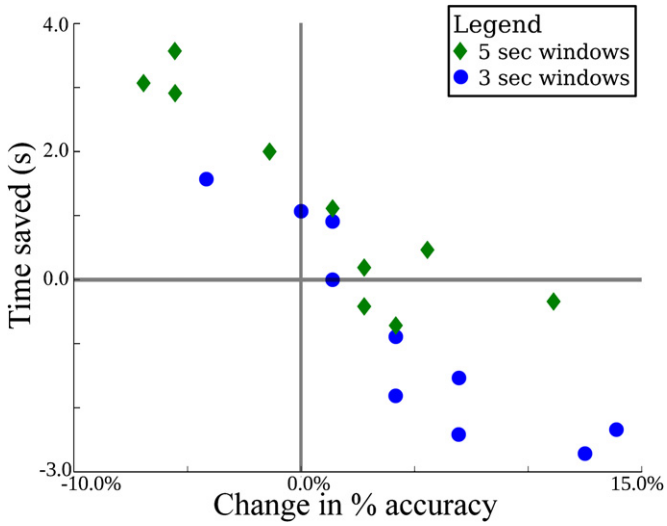


Figure 6. Change in accuracy and average decision time for each user for the POMDP BCI, relative to the baseline methods (fixed-size time window Gaussian classifiers).

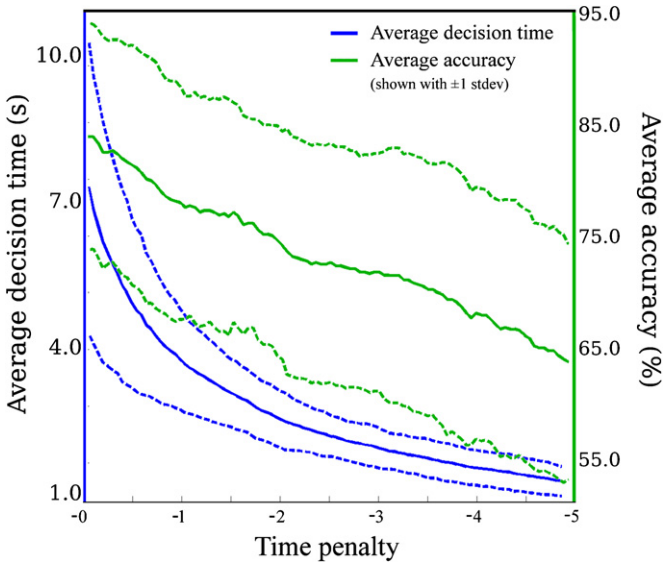


Figure 7. Average time taken to make a control decision and average accuracy as a function of the time penalty parameter r_{wait}, averaged over all users. Dashed lines denote ± 1 standard deviation.

Overall, we found that with the chosen parametrization of the POMDP model, all users saw either an increase in accuracy, a decrease in decision time, or both compared to Gaussian classifiers with fixed-size time windows (figure 6, table 2). We emphasize, however, that in practice one would choose a parametrization specific to each user that reflects their personal preferences in terms of speed and accuracy for a particular task. In the next section, we explore this trade-off empirically.

5.2. Effect of reward function on decision time

We now show how the POMDP BCI can capture the preferences of users with regard to speed versus accuracy using the reward function. Suppose the user values speed over accuracy in a particular task. We can express this preference by making the parameter r_{wait} high relative to the cost of a poor control decision. On the other hand, if accuracy is critical, r_{wait} can be set to a lower value, allowing the POMDP to make more careful decisions.

To test this hypothesis, we ran the same offline experiments as above on all users but varied the time penalty. As seen in figure 7, the POMDP BCI exhibits the desired behaviour in terms of speed versus accuracy: when the time penalty in the POMDP model is increased, the BCI tends to make faster, but generally less accurate decisions and vice versa.

5.3. Adaptation to the user's signal-to-noise ratio

The optimal POMDP policy depends not only on the reward function but also on the observation model for the specific user. The observation model is directly related to the value of the information the BCI could collect by waiting. If the observations (brain signals) for a user are particularly noisy, the POMDP model will recognize that collecting additional information will be comparatively less useful. As a result, it will lower the confidence threshold at which it is willing to make a decision to avoid some of the time penalty for waiting.

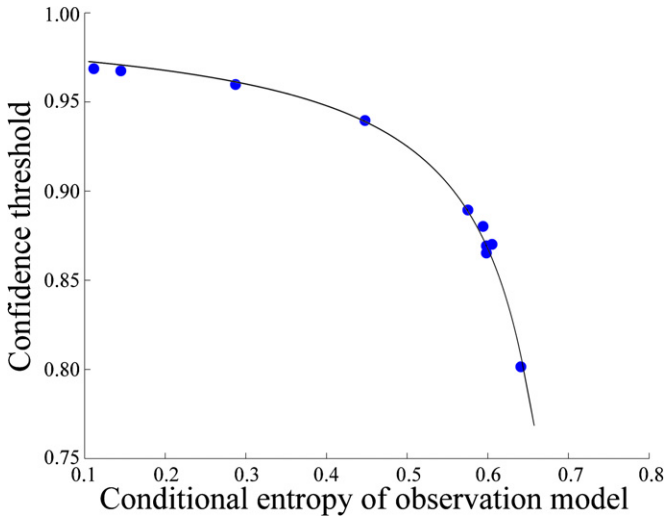


Figure 8. Conditional entropy of the observation model versus confidence threshold at which a control decision is made, shown for all users.

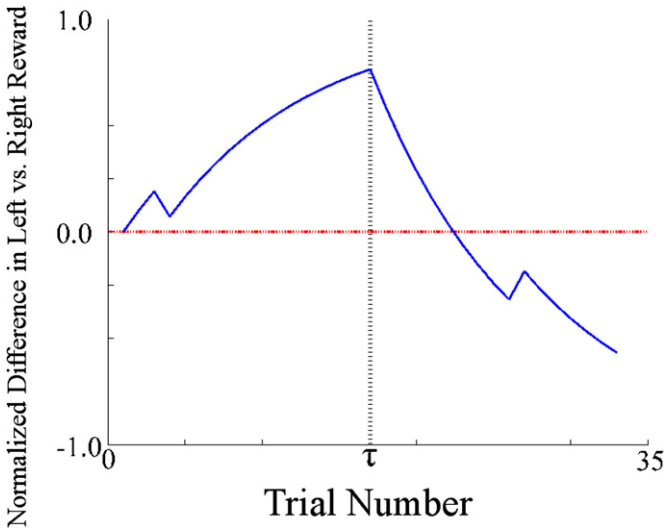


Figure 9. Normalized difference in expected reward for a control mapping for user 1 as a function of time. The curve shows the BCI's preference for choosing action a_{left} versus a_{right} while in state 1. Note that we show only one curve for the two states since they mirror each other, i.e., as one state gets mapped to a_{left}, the other gets mapped to a_{right} in precisely the same way. At trial no. τ, the user switches control mappings and the system begins to converge to the other mapping, as indicated by the downward trend of the curve.

To quantify this effect, we calculated a conditional entropy measure from the users' observation models and explored its relationship with the POMDP BCI's decision threshold for the SSVEP task. Our conditional entropy measure specifies the expected entropy of the brain states given an observation, measuring how useful we expect the observation to be in conveying information about the user's brain state. Formally, the measure is defined as:

$$H(S|O) = \sum_{o \in O} p(o) H(S|O = o)$$
$$= \sum_{o \in O} p(o) \sum_{s \in S} p(s|o) \log \frac{1}{p(s|o)}. \quad (13)$$

We compute $p(s|o)$ and $p(o)$ from the POMDP observation model $p(o|s)$ using Bayes' rule and a uniform prior $p(s)$ over states.

Figure 8 shows that this measure of noise in the user's observations is directly related to the confidence threshold for each user computed by the POMDP's optimal policy. The curve fit to the data shown in the plot captures the regularity we saw amongst our subjects (because we had only ten data points, we refrained from performing any hypothesis tests but we note that the R^2 measure in this case was 0.99).

Figure 8 provides some useful insights into the POMDP BCI's behaviour. For instance, if the threshold was set equally high for all users, it would imply that a user with a less discriminable signal might have to wait an unreasonable amount of time for a decision. We tested this hypothesis by investigating how long user 9 (the user with the noisiest signal) would have to wait to get decisions as confident as user 1 (who had the strongest signal). In many cases, user 9 would have had to wait more than 12 s for a BCI decision, far too long for a usable interface.

5.4. Co-adaptation in the POMDP BCI

We found that in the closed-loop experiments, the POMDP BCI successfully discovered each user's initial control mappings, and also detected subsequent changes to the mappings when the user decided to switch.

5.4.1. Co-adaptive experiment 1—the simple case. Figure 9 shows an example of the system's performance in this first experiment. The plot shows the difference in expected reward between the two possible control mappings as a function of time. The system picks the mapping with the highest expected reward. As a result, the BCI uses one mapping when it is above the red line in the plot, and the other when it is below.

At the beginning of the experiment, the BCI has no bias towards one mapping or another, and therefore the difference in expected reward is 0. As the trials proceed, the BCI gradually converges to the correct mapping, as indicated by the increase in expected reward (figure 9). When the user switches control mappings at time τ, the BCI begins receiving negative feedback, which causes it to eventually switch to the other control mapping (when it crosses the red dashed line). Table 3 summarizes the performance of the co-adaptive BCI across all users.

5.4.2. Co-adaptive experiment 2—user exploring inputs. Our 5-channel experiment with three users indicates that the co-adaptive BCI can cope with a larger number of brain states (here, SSVEP channels). We allowed the users to explore which channels seemed to work best for them, and the BCI system generally adapted to the exploration. Figure 10 summarizes the results by user.

5.4.3. Co-adaptive experiment 3—exploration by the BCI. We conducted this experiment in three phases. In the first phase, the user chose an initial control mapping; in each of

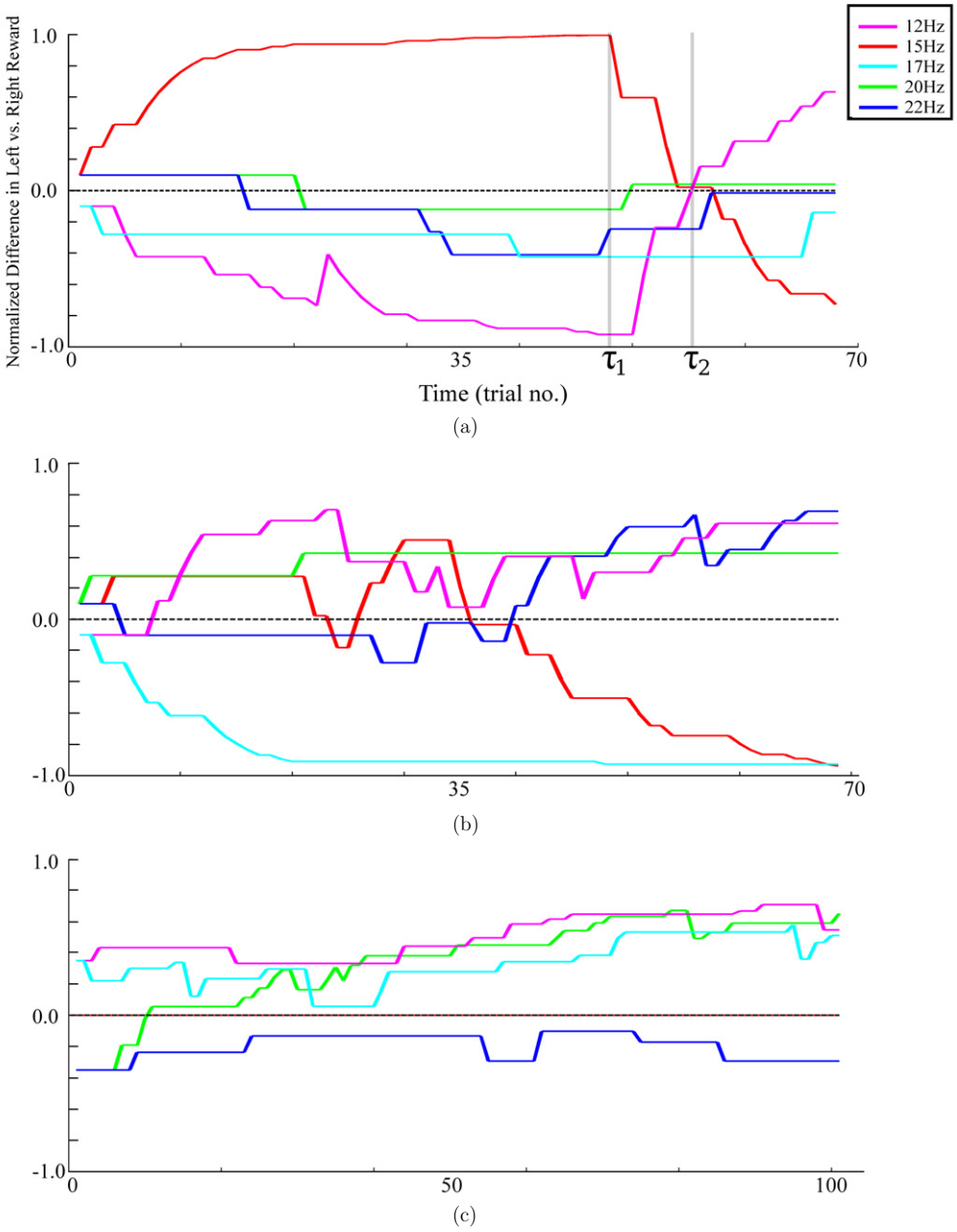


Figure 10. Co-adaptive convergence in the 5-channel experiment. Each curve shows the BCI's preference for choosing action a_{left} versus a_{right} while in a given state. (a) User 1: this user initially attempted to use the 17 Hz and 15 Hz channels for control, followed by the 22 Hz and 15 Hz channels. Ultimately, the 12 Hz and 15 Hz appeared to work best in terms of accuracy. At trial no. τ_1, the user's performance with these two frequencies reaches maximum separation. The user then reverses the mapping; the reversal was completed at τ_2 when the 12 Hz and 15 Hz channels cross the zero point in opposite directions. (b) User 2: the user found early on that 17 Hz worked well for one target, but required some exploration before deciding on 12 Hz and 22 Hz for the other target. (c) User 3: this user had very high noise during the experimental session, but still managed to find a consistent mapping. The slow convergence was due to the classifier's lack of confidence (see the role of the confidence parameter β in equation (10)).

the subsequent two phases, the user changed their control mapping. We found our exploration-based BCI successfully identified the user's intended mapping in all cases, except in the final phase for user 4 in which, though convergence was occurring, user fatigue caused us to stop the experiment early.

As seen in figure 11(a), user 1's mappings were easily identified. In phase 1 (black arrows), state 1 (12 Hz channel) was mapped by the POMDP BCI to action 3 (choosing the centre target), state 2 (15 Hz channel) to action 1, etc. In each subsequent phase, when some change occurred, the system was able to adapt to the change (depicted by the red arrows and the blue arrows). User 2 had somewhat noisier control but still successfully completed the task (figure 11(b)). User 3 was mostly successful in achieving co-adaptive control (figure 11(c)), though we stopped the experiment in phase 3 due to subject fatigue. Overall, the results suggest that the POMDP BCI can successfully identify the users' changes in control strategy as they occur.

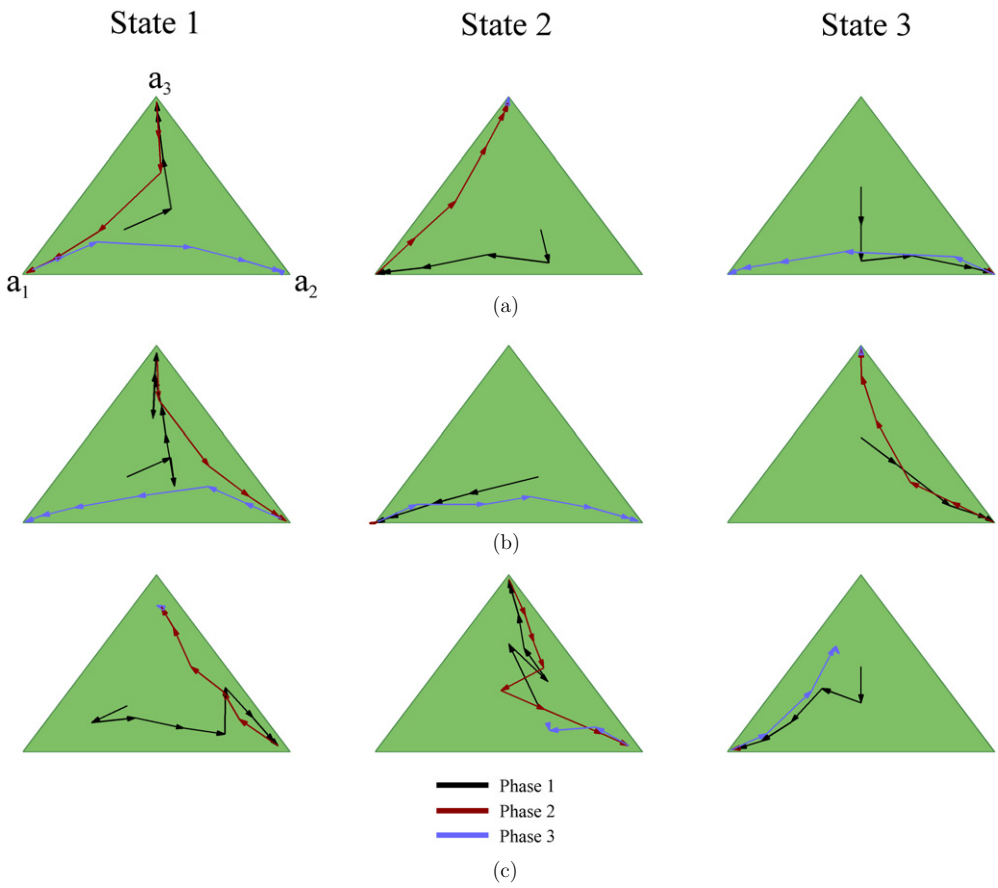


Figure 11. Co-adaptive convergence With BCI-based exploration. Each graph is a simplex over our action selection probability distributions. A point in each graph represents a probability distribution over the three possible actions the system can take while in that state. States 1, 2, 3 refer to the 12 Hz, 15 Hz, and 17 Hz channels respectively. Actions a_1, a_2 and a_3 refer to choosing the left, right, and centre targets respectively. As learning occurs, the probability distributions change, indicated by trajectories in the space. (a) User 1. (b) User 2. (c) User 3 (did not finish phase 3).

Table 3. Performance of the simple co-adaptive BCI across all users. The table shows the percentage of trials in which the co-adaptive system found the correct control mapping. For comparison, we show the accuracy of the offline POMDP experiment from above.

User ID	% of trials with correct control mapping[a]	Offline POMDP accuracy
1	100.0	94.4
2	97.0	90.2
3	80.3	80.5
4	90.9	90.2
10	62.1	90.2

[a] We defined 'correct' acknowledging that it takes the system some amount of time to switch control mappings. Thus, we assume the system should have the correct control mapping before τ, and the correct mapping after τ, less the amount of time required for the fastest possible reversing of the control mapping. The fastest possible reversing occurs when the classifier has 100 per cent accuracy and confidence.

6. Comparison with related work

Our proposed approach to BCI design differs from traditional BCI approaches in several ways. First, most existing approaches, including our own past work, use a fixed-sized time window for classification (e.g., SSVEP BCI: [31], P300 BCI: [9], imagery-based BCI: [33]) rather than automatically tailoring the decision time window for each individual user as in the proposed POMDP model. Because of differences between users and the different circumstances in which a BCI may be used, we feel a user-adaptive decision time window is important for real-world applications.

Park *et al* [18] were the first, to our knowledge, to utilize a POMDP in the context of a BCI, focusing specifically on the problem of P300 detection. They proposed using a POMDP for deciding which rows and columns to flash in a P300 speller application, with the goal of decoding the user's intention more efficiently. As in our approach, they use a POMDP to balance the trade-off between accuracy and speed. However, their method is intrinsically tied to a specific control paradigm (the P300 speller) and therefore does not readily generalize to other paradigms (e.g., SSVEP [20] systems where the stimulus is fixed and motor-imagery-based [34] systems where the user response is self-paced). Additionally, their model does not allow co-adaptation based on reinforcement from the environment, an important goal of our approach.

Vidaurre *et al* [14] and DiGiovanna *et al* [15] have both proposed co-adaptive BCI systems but based on different methods. Vidaurre *et al* approach used reinforcements to refine

subject-specific classifiers and spatial filters. This created BCIs which would work for some users for whom previous systems lacked reasonable accuracy (a phenomenon that has been called 'BCI illiteracy'). DiGiovanna *et al* proposed a co-adaptive system that mapped the brain signal of a rat to the movement of a prosthetic device, which produced water reward for the rat. Our approach differs from these earlier co-adaptive BCIs in that we explicitly model uncertainty in the brain (and environment) state, modulating the BCI's behaviour in a context- and user-specific manner. This is important because the signal processing and control problems in real-world BCIs are inherently noisy. Due to the need for risk aversion in critical tasks, the varying uses of a BCI, and the varying discriminability of users' brain signals, we believe that an approach to making decisions based on uncertainty is necessary for real-world BCI applications.

7. Discussion and future work

Our results suggest that the framework of POMDPs offers a promising approach for designing general-purpose BCIs that can handle uncertainty in neural signals and co-adapt with the user on an ongoing basis. We demonstrated the approach using a simple SSVEP-based BCI, but the basic POMDP framework is general enough to be applicable to other EEG paradigms such as imagery-based control and invasive BCI paradigms based on intracortical or electrocorticographic recordings. The complexity of the POMDP methods needed for higher degrees-of-freedom neural control problems is still relatively small compared to recent AI problems tackled using POMDPs, some of which have several thousands of states (see [25] for examples). We therefore believe that the POMDP approach will scale well to more sophisticated neural engineering applications such as controlling prosthetic devices.

Our results also demonstrate how the co-adaptive capability of the POMDP BCI offers the user the ability to choose the control scheme she/he finds most intuitive for a given situation, allowing them to, in effect, search the space of available 'brain states'. The BCI relies on reinforcements (penalties and rewards) to automatically converge to the appropriate actions for the user's chosen mapping, assuming the user continues to use the particular choice for several trials. We also showed how the BCI allows the user's control strategy to be *non-stationary* over time, automatically co-adapting with the user to a new strategy when the user decides to change the mapping.

The fact that the POMDP BCI maintains a *belief* (posterior probability distribution) over the user's brain state allows a much more powerful form of decision making than traditional BCI approaches, which have typically been based on the output of classifiers or regression techniques. By computing actions based on the entire probability distribution over brain states, the BCI is able to make decisions in a flexible manner according to the amount of uncertainty currently being experienced regarding the user's estimated brain state. We showed how this results in a speed–accuracy trade-off dictated by how noisy each individual user's SSVEP signal is. The trade-off can in turn be controlled in a principled manner via the reward function, which expresses the preferences of the user as well as the BCI application (e.g., high priority for safety in prosthetics or wheelchair control applications).

A related advantage of POMDPs is that they allow us to model uncertainty over not just brain state but also the environment. This becomes important in a real-world deployment of a BCI because complex interactions with the user's environment will typically involve imperfect modelling. For instance, suppose a BCI user commands a prosthetic arm to 'pick up the leftmost object on the table' but that the robot has an unreliable model of the objects in front of it. Even if the BCI is very confident about the user's intention, it may need to wait and collect more information from the robot's sensors about the environment state before acting. The POMDP framework lends itself naturally to this type of probabilistic reasoning and decision making. We are presently exploring this approach to joint modelling of brain/environmental uncertainty using a BCI-controlled robot application.

One aspect of co-adaptive control not explored in this paper is the non-stationarity in the observation model, both within a trial (due to fast adaptive processes in the brain) as well as across trials (due to impendence changes in electrodes, changes in brain activation levels, etc). For example, during our experiments, we found that the discriminability of the users' brain signals was highest between 1–3 s into the trial for all users except user 3. Because this is a predictable non-stationarity, we could learn a time-varying observation model, enabling the POMDP to make more accurate decisions by not waiting for less reliable late-trial information. We hope to explore this and other ideas for incorporating non-stationary observation models in future work.

Another aspect of co-adaptive control not explored in this paper is the use of novel forms of feedback. In our case, feedback occurs after an incorrect selection has been made, where 'correct' is defined in a task-specific way. However, feedback can also come from a number of other sources which act prospectively to prevent errors (e.g., an impending collision signal from sensors on an intelligent wheelchair), rather than diagnostically after an error has already occurred. In the future, we intend to further explore these and other novel sources of reinforcement (e.g., brain signals such as error potentials [16]) for co-adaptive control in a POMDP BCI.

Acknowledgments

This research was supported by Army Research Office (ARO) award no. W911NF-11-1-0307, the Office of Naval Research (ONR) grant N000140910097, NSF award no. 0930908, NSF Center for Sensorimotor Neural Engineering (EEC-1028725), and a Mary Gates Research Scholarship awarded to Matthew Bryan. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Science Foundation or the other funding agencies.

References

[1] Rao R P N 2013 *Brain–Computer Interfacing: An Introduction* (Cambridge: Cambridge University Press)

[2] Wolpaw J R, Birbaumerc N, McFarland D J, Pfurtscheller G and Vaughan T M 2002 Brain–computer interfaces for communication and control *Clin. Neurophysiol.* **113** 767–91
[3] Dornhege G, Millán J R, Hinterberger T, McFarland D J and Müller K-R 2007 *Towards Brain–Computer Interfacing* (Cambridge, MA: MIT Press)
[4] Wessberg J, Stambaugh C R, Kralik J D, Beck P D, Laubach M, Chapin J K, Kim J, Biggs S J, Srinivasan M A and Nicolelis M A 2000 Real-time prediction of hand trajectory by ensembles of cortical neurons in primates *Nature* **408** 361–5
[5] Velliste M, Perel S, Spalding M C, Whitford A S and Schwartz A B 2008 Cortical control of a prosthetic arm for self-feeding *Nature* **453** 1098–101
[6] Serruya M D, Hatsopoulos N G, Paninski L, Fellows M R and Donoghue J P 2002 Brain–machine interface—instant neural control of a movement signal *Nature* **416** 141–2
[7] Farwell L A and Donchin E 1988 Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials *Electroencephalogr. Clin. Neurophysiol.* **70** 510–23
[8] Bensch M, Karim A A, Mellinger J, Hinterberger T, Tangermann M, Bogdan M, Rosenstiel W and Birbaumer N 2007 Nessi: an EEG-controlled web browser for severely paralyzed patients *Comput. Intell. Neurosci.* **2007** 71863
[9] Bell C J, Shenoy P, Chalodhorn R and Rao R P N 2008 Control of a humanoid robot by a noninvasive brain–computer interface in humans *J. Neural Eng.* **5** 214
[10] Galán F, Nuttin M, Lew E, Ferrez P, Vanacker G, Philips J and Millán J del R 2008 A brain-actuated wheelchair: asynchronous and non-invasive brain–computer interfaces for continuous control of robots *Clin. Neurophysiol.* **119** 2159–69
[11] Millan J R, Renkens F, Mouriño J and Gerstner W 2004 Noninvasive brain-actuated control of a mobile robot by human EEG *IEEE Trans. Biomed. Eng.* **51** 1026–33
[12] Rao R P N and Scherer R 2010 Statistical pattern recognition and machine learning in brain–computer interfaces *Statistical Signal Processing for Neuroscience and Neurotechnology* (Amsterdam: Elsevier) pp 335–68
[13] Gürel T and Mehring C 2012 Unsupervised adaptation of brain machine interface decoders arXiv:1206.3666
[14] Vidaurre C, Sannelli C, Müller K R and Blankertz B 2011 Co-adaptive calibration to improve BCI efficiency *J. Neural Eng.* **8** 025009
[15] DiGiovanna J, Mahmoudi B, Fortes J, Principe J C and Sanchez J C 2009 Coadaptive brain–machine interface via reinforcement learning *IEEE Trans. Biomed. Eng.* **56** 54–64
[16] Buttfield A, Ferrez P W and Millan J 2006 Towards a robust BCI: error potentials and online learning *IEEE Trans. Neural Syst. Rehabil. Eng.* **14** 164–8
[17] Kaelbling L P, Littman M L and Cassandra A R 1998 Planning and acting in partially observable stochastic domains *Artif. Intell.* **101** 99–134
[18] Park J, Kim K E and Song Y K 2011 A POMDP-based optimal control of P300-based brain–computer interfaces *AAAI'11: Proc. Association for the Advancement of Artificial Intelligence Conf. (AAAI)* pp 1559–62
[19] Sutton R S and Barto A G 1998 *Reinforcement Learning: An Introduction* (Cambridge, MA: MIT Press)
[20] Müller-Putz G R and Pfurtscheller G 2007 Control of an electrical prosthesis with an SSVEP-based BCI *IEEE Trans. Biomed. Eng.* **55** 361–4
[21] Middendorf M, McMillan G, Calhoun G and Jones K S 2000 Brain–computer interfaces based on the steady-state visual-evoked response *IEEE Trans. Rehabil. Eng.* **8** 211–4
[22] Cheng M, Gao X, Gao S and Xu D 2002 Design and implementation of a brain–computer interface with high transfer rates *IEEE Trans. Biomed. Eng.* **49** 1181–6
[23] Brunskill E, Kaelbling L, Lozano-perez T and Roy N 2008 Continuous-state POMDPs with hybrid dynamics *Symp. on Artificial Intelligence and Mathematics* pp 13–18
[24] Hoey J and Poupart P 2005 Solving POMDPs with continuous or large discrete observation spaces *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)* pp 1332–8
[25] Kurniawati H, Hsu D and Lee W S 2008 SARSOP: efficient point-based POMDP planning by approximating optimally reachable belief spaces *Proc. Robotics: Science and Systems IV (Zurich, Switzerland)* www.roboticsproceedings.org/rss04/p9.html
[26] Smith T and Simmons R 2005 Point-based POMDP algorithms: improved analysis and implementation *Proc. Conf. on Uncertainty in Artificial Intelligence* pp 542–7
[27] Spaan M T J and Vlassis N 2005 Perseus: randomized point-based value iteration for POMDPs *J. Artif. Intell. Res.* **24** 195–220
[28] Hsu D *et al* 2012 Motion modelling, analysis, and planning project http://bigbird.comp.nus.edu.sg/pmwiki/farm/motion/index.php?n=Site.PomdpPlanning
[29] Chung M, Bryan M, Cheung W, Scherer R and Rao R P N 2011 Interactive hierarchical brain–computer interfacing: uncertainty-based interaction between humans and robots *BCI'11: 5th Int. Brain–Computer Interface Conf. (Graz, Austria)* pp 20–3
[30] Abou-Moustafa K T, Torre F and Ferrie F P 2010 Designing a metric for the difference between Gaussian densities *Brain, Body and Machine Advances in Intelligent and Soft Computing* vol 83 ed J Angeles, B Boulet, J J Clark, J Kövecses and K Siddiqi (Berlin: Springer) pp 57–70
[31] Bryan M, Nicoll G, Thomas V, Chung M, Smith J R and Rao R P N 2012 Automatic extraction of command hierarchies for adaptive brain–robot interfacing *ICRA'12: IEEE Int. Conf. on Robotics and Automation (ICRA)* pp 3691–7
[32] Bryan M J, Green J, Chung M, Chang L, Scherer R, Smith J and Rao R P N 2011 An adaptive brain–computer interface for humanoid robot control *11th IEEE-RAS Int. Conf. on Humanoids Robots* (Piscataway, NJ: IEEE) pp 199–204
[33] Fabiani G E, McFarland D J, Wolpaw J R and Pfurtscheller G 2004 Conversion of EEG activity into cursor movement by a brain–computer interface (BCI) *IEEE Trans. Neural Syst. Rehabil. Eng.* **12** 331–8
[34] Pfurtscheller G, Neuper C, Flotzinger D and Pregenzer M 1997 EEG-based discrimination between imagination of right and left hand movement *Electroencephalogr. Clin. Neurophysiol.* **103** 642–51