A Low-Complexity Brain–Computer Interface for High-Complexity Robot Swarm Control

Gregory Canal®, Yancy Diaz-Mercado®, Member, IEEE, Magnus Egerstedt®, Fellow, IEEE, and Christopher Rozell®, Senior Member, IEEE

Abstract—A brain-computer interface (BCI) is a system that allows a human operator to use only mental commands in controlling end effectors that interact with the world around them. Such a system consists of a measurement device to record the human user’s brain activity, which is then processed into commands that drive a system end effector. BCIs involve either invasive measurements which allow for high-complexity control but are generally infeasible, or noninvasive measurements which offer lower quality signals but are more practical to use. In general, BCI systems have not been developed that efficiently, robustly, and scalably perform high-complexity control while retaining the practicality of noninvasive measurements. Here we leverage recent results from feedback information theory to fill this gap by modeling BCIs as a communications system and deploying a human-implementable interaction algorithm for noninvasive control of a high-complexity robot swarm. We construct a scalable dictionary of robotic behaviors that can be searched simply and efficiently by a BCI user, as we demonstrate through a large-scale user study testing the feasibility of our interaction algorithm, a user test of the full BCI system on (virtual and real) robot swarms, and simulations that verify our results against theoretical models. Our results provide a proof of concept for how a large class of high-complexity effectors (even beyond robotics) can be effectively controlled by a BCI system with low-complexity and noisy inputs.

Index Terms—Brain-computer interface, feedback coding, information theory, swarm robotics.

BRAIN-computer interfaces (BCI) are systems that consist of hardware to measure a human user’s brain activity, an interaction algorithm to map the user’s mental commands to control signals, and an end effector that the user operates via these control signals. This direct link between brain and effector provides a means for paralyzed users to circumvent muscular pathways and interact with everyday devices [1] as well as an augmented interface for healthy users. There are several tradeoffs involved in the design of BCIs, including whether measurements are taken invasively or noninvasively, how many mental commands are needed to drive the effector to a desired behavior, how scalable the system is to effectors of varying complexity, and how robust the system is to user error and system noise in measurement processing. Although BCIs with invasive neural measurements have had experimental success in controlling high-complexity effectors (e.g., robotic arms [2], [3], [4], [5]) with many degrees of freedom, such BCIs are only available in research settings and require a surgical procedure for electrode implantation. BCIs with noninvasive measurements (e.g., scalp electrode recordings via an electroencephalogram (EEG)) are more widely implementable due to their relative ease of use and lower cost, but are limited to controlling comparatively simpler effectors (e.g., basic wheelchair control [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], cursor control [18], [19], [20], [21], [22], [23]) with few degrees of freedom due to lower signal-to-noise ratios. In recent years there has been an emerging interest in improving these tradeoffs for neurotechnology in commercial and clinical applications, with aims to both broaden intended uses and engineer higher quality BCI devices [24], [25], [26], [27].

Despite this increased interest, there remains a large gap between the complexity of potential end effectors and the capabilities of interaction algorithms that map the user’s mental commands from noninvasive interfaces to control signals. There are several specifications required for a noninvasive interaction algorithm to meet this need. First and foremost, any such algorithm must be implementable by humans through mental commands easily learned with training. Furthermore, such interaction algorithms must be scalable so that they remain tractable with a minimal increase in user overhead when controlling more complex effectors. Similarly, increases in effector complexity should not result in a need for increased measurement capabilities (e.g., more EEG features). Because even the most advanced...
BCI measurements are susceptible to errors, an interaction algorithm must be robust to such errors. Finally, due to the wide range of applications that can benefit from BCIs, an interaction algorithm should be designed for general use and be adaptable to a variety of specific tasks.

Currently, interaction algorithms for noninvasive BCIs fall into two broad categories that only achieve a subset of these specifications. In the first category, the user selects discrete effector behaviors from a finite set of options displayed on an interface, such as choosing waypoints for a motorized wheelchair [17] or selecting letters on a virtual keyboard [28], [29], [30], [31]. Although this type of interaction is easy to use, it scales poorly since it becomes increasingly tedious for a user to select their desired behavior as the number of options (i.e., the precision) increases. In the second category, continuous features from measured brain activity are directly mapped to continuous control over effector action spaces with arbitrary precision (e.g., robotic arm control [32], [33], quadcopter control [34], cursor control in up to three-dimensional space [18], [19], [20], [21], [22], [23]). Unlike discrete selection, continuous control allows a user to navigate an effector’s action space with arbitrary precision in a scalable manner. However, this type of interaction is severely limited in that each additional effector degree of freedom requires an independent, continuous measurement feature, which scales poorly and typically limits an effector to at most three degrees of freedom for EEG-based BCIs.

The main contribution of this paper is a robust interaction algorithm that reaps the benefits of both discrete selection and continuous control while addressing the critical disadvantages of each. The key innovation of our information-theoretic approach is that each new input is used in conjunction with closed-loop feedback to the user to efficiently refine the entire effector state simultaneously through a sequence of simple and tractable decisions. While our approach builds off of techniques used in prior work [35], [36], [37], [38], it utilizes a new method for effector parameterization that significantly expands the class of controllable systems. We test our approach on human control of a mobile robot swarm (a large collection of robots, as depicted in Fig. 1c), where a human operator issues high-level, global commands which are executed by the swarm in a distributed fashion (individual robot depicted in Fig. 1d).

Robot swarm control serves as an ideal testbed for our approach, since robot swarms are high-complexity cyber-physical systems that can be naturally parameterized beyond three degrees of freedom and have been previously tested in a BCI setting [39]. Part of what makes robot swarm control complex is the necessity to coordinate the individual robot motion to avoid collisions while attempting to achieve their objectives, e.g., reach a target formation. Robot swarms typically consist of weak robots which possess limited computation, sensing, and communication capabilities. Thus, in order to achieve the desired behavior, the control must rely on local sensing information and scale well in complexity with the number of robots in the swarm. These local rules result in the desired global emergent behavior. When humans are involved, the swarm formation must be achieved quickly and be cohesive enough to provide the human operator with clear visual feedback to aid in the decision-making.

Over the last couple of decades, there have been many developments in large classes of coordination algorithms and abstractions that support the required mapping from low-complexity, high-level commands to highly complex coordinated swarm behaviors [40]. Recent advances in coverage control [41], [42] provide an excellent approach to perform this mapping for formation control. The algorithms allow for a human operator to broadcast reference swarm spatial densities and boundaries in the robot domain that encode desired formations. The robots in the domain can then coordinate their motion with nearby robots to robustly achieve the commanded density distributions in real time in a scalable, distributed manner. In this paper, we show through an array of human trials and simulations that refining the entire state space is an effective approach for BCI swarm control, demonstrating the potential and flexibility of our method for controlling high-complexity end effectors with low-complexity inputs.

II. METHODS

A. Algorithm for Refining End Effector Behavior

To understand our interaction algorithm at a high-level, first consider the task of finding a word in an English dictionary. A natural strategy is for the user to repeatedly bisect the remaining pages depending on whether their desired word comes before or after the current page. Our interaction algorithm is analogous to this efficient search procedure: the BCI user selects an effector behavior from an ordered dictionary of candidate behaviors through a sequence of bisections. Specifically, suppose that the BCI user learns a lexicographical ordering rule for the set of effector behaviors, which determines a total order of behaviors organized as a dictionary. At each round of interaction, the effector presents to the user the behavior that bisects the remainder of the dictionary. The user indicates to the effector (via a binary mental command) if their desired behavior precedes or succeeds the candidate behavior, and the dictionary scope is narrowed based on their reply. Rather than strict elimination of half of the dictionary at each step, the algorithm uses a probabilistic weighting over the dictionary to account for possible noise in the user’s input. Eventually, the user will have provided enough refinements for the end effector to correctly converge to the user’s desired behavior. Importantly, this procedure does not involve the adjustment of individual effector parameters, but instead only requires the user to decide on the precedence of their total desired behavior with respect to the current candidate behavior. Although each bisection affects every effector parameter, the user only has to make a simple binary decision at each round, regardless of the number of parameters; this is distinct from a brute-force approach where the user adjusts each parameter individually.

While this interaction algorithm is intuitively satisfying, it is also endowed with rigorous performance guarantees that become apparent when the entire interface is framed as a feedback communications system: the human user acts as
Refining effector behavior through configuration sorting. Effector behavior is determined through iterative refinement from the BCI user. (a) In the example of robot swarm configuration refinement, the BCI user indicates through a mental command (e.g., binary motor imagery) if their desired configuration comes before or after the current configuration in the swarm dictionary (see (b)). A computer decodes the input by classifying scalp electrode recordings from the user, and updates a posterior distribution over the configuration dictionary. The median of the updated distribution is selected as a new configuration guess and transmitted to the swarm through a global update. Each individual robot then adjusts its position locally so that the overall configuration conforms to the new guess in a distributed manner. (b) In the swarm configuration dictionary, character alphabets are defined in order from first to last as follows, with alphabet precedence in parentheses: horizontal position of the configuration center (centers to the left preceding centers to the right); vertical position of the configuration center (centers below preceding centers above); number of sides (fewer sides preceding more sides); configuration size as the radius from the center to each vertex (smaller radii preceding larger radii). Each example panel depicts a pair of strings whose critical character corresponds to the panel column, with blue (solid) configurations preceding red (dashed) configurations in the overall dictionary ordering. (c) Example of a robot swarm coordinating to form a globally specified configuration by using only local information. (d) Close-up view of an individual mobile robot used in our demonstrations. A robot swarm (as in (c)) consists of several such robots collectively performing global actions in a distributed manner.

A “transmitter” by encoding their desired effector behavior (the “message”) through a sequence of binary BCI inputs (“codes”). These inputs are sequentially decoded by the end effector to refine a new estimate of the user’s desired behavior, which is fully observable to the user as “noiseless feedback” and informs the choice of their next input. Because there is some chance that the user’s binary input will be misclassified or that the user will make a decision error, the sequence of classification results can be modeled as outputs of a noisy binary symmetric channel (BSC) with a crossover probability equal to the misclassification probability. When framed as such a communications system, our interaction algorithm is mathematically equivalent to the posterior matching coding scheme [43]. Posterior matching is an optimal capacity-achieving code [44], meaning that this interaction algorithm communicates the user’s desired behavior to the effector with as few binary inputs as possible for a given error rate.

In previous work, posterior matching has been used as an interaction algorithm in noninvasive BCIs for tasks such as text entry or vehicle path planning [35], [36], [37], [38]. In these cases, a dictionary of ordered effector behaviors can be formed by constructing each dictionary element, or string, as a concatenation of characters from a fixed alphabet. For example, in text entry and path planning a string is constructed as a concatenation of English language letters and arc segments, respectively. In either of these cases, the precedence between two strings can be determined by identifying the first character that differs between the strings (referred to here as the critical character), and assigning precedence to the string whose critical character comes earliest in the character alphabet (e.g., ‘a’ precedes ‘z,’ arcs angled left precede arcs angled right). We refer to such dictionaries as homogeneous since in each case a single alphabet is used for all character positions in the behavior string. Tasks like text entry or path planning can be adequately modeled by homogeneous dictionaries, since each additional effector parameter (e.g., letter, segment) is of the same type.

Unlike the tasks described above, many high-complexity effectors cannot be described with homogeneous dictionaries by concatenating characters from a single alphabet. For example, in robot swarm control, each swarm configuration is
characterized by varied parameters describing position, shape, and size. To model these high-complexity effectors we design a heterogeneous dictionary, where a different alphabet is used for each character position in the behavior string. To our knowledge, posterior matching has not been deployed as an interaction algorithm using heterogeneous dictionaries, and it was previously unknown if BCI users can successfully learn and apply a heterogeneous dictionary to posterior matching control of a high-complexity effector. As we detail below, we demonstrate in a large-scale interface study that people can learn such a heterogeneous dictionary with little training and make pairwise string comparisons with high proficiency.

While one might conceive of a variety of heterogeneous dictionaries to describe swarm configurations, here we adopt a dictionary of regular polygons as a proof of concept. Each polygon string is parameterized by characters including horizontal position, vertical position, number of sides, and size, with distinct alphabets for each character position (Fig. 1b). To search this polygon dictionary with posterior matching, the BCI user issues hand motor imagery (MI) inputs detected via EEG measurements to indicate if the desired behavior comes before or after the currently demonstrated behavior in the dictionary. MI tasks are a well-studied and popular binary input modality where the user mentally visualizes wrist flexions of either their left or right hand and the resulting changes in EEG frequencies are detected by a binary classifier [45], [46].

To refine the swarm, the user determines the first character where their desired configuration differs from the current configuration and issues a left-hand (right-hand) MI input if their desired polygon preceded (succeeds) the current polygon at the critical character. As the complexity of the dictionary increases, the sequential scan to find the critical character may take marginally more time, but the decision by the user is ultimately based only on a simple evaluation of that character (despite each user input potentially updating all characters). Note that this approach is not limited to EEG-based MI, and is compatible with any binary input mechanism including inputs detected by invasive BCIs. We refer to this combination of a heterogeneous swarm dictionary with binary input posterior matching as SCINET: Swarm Control via Interactive Neural Teleoperation (illustrated in Fig. 1a).

As detailed in Supplement B.7, this procedure is formally modeled by mapping each string to a point on the unit interval [0, 1) such that one string precedes another in the dictionary if and only if its real number representation is less than that of the other string. As each binary motor imagery input is issued, a posterior distribution is updated over the unit interval according to Bayes’ rule, tracking where the target string’s corresponding point is most likely to lie. The effector’s estimated string is selected at each interaction round by computing the median of this posterior distribution, and mapping this point back to its corresponding string. The posterior median is selected as the effector’s estimate since it divides the posterior distribution in half, which is the probabilistic equivalent of performing a hard bisection.

1) Dictionary Construction: We constructed the swarm dictionary with the following characters in each configuration string, in order of character precedence: horizontal position, vertical position, number of sides, and size. Horizontal position and vertical position refer to the coordinates of the center of each polygon, respectively (Supplement Fig. 1). Size refers to the distance between the polygon center and each vertex (this value is the same for each vertex since the polygons are regular). The number of characters in each alphabet is as follows: 5 horizontal positions; 2 vertical positions; 3 numbers of sides; and 2 polygon sizes. Characters in the horizontal position alphabet were chosen to uniformly span the robot arena (virtual or physical), as were the characters in the vertical position alphabet. The “number of sides” alphabet has characters given by 3, 4, or 5 sides, with the polygon rotation set by fixing a vertex at the “12 o’clock” position of each shape. The two size distances were tuned such that size differences were visually discernible, while not causing robots to overflow outside the span of the arena. With an arena of width 1.5 and height 1 (specified in units relative to the arena height), these specifications translate to the following alphabet characters: horizontal position [0.4, 0.575, 0.75, 0.925, 1.10]; vertical position [0.4, 0.6]; number of sides [3, 4, 5]; size [0.3, 0.4]. These values (except for number of sides) are specified in abstract units relative to the arena height, and are scaled at runtime to the physical dimensions of the actual swarm arena; for instance, if the physical swarm arena is 2.5 feet in height, then the first horizontal position character is 0.4 × 2.5 = 1 foot from the left arena edge. In total, this combination of alphabets produces a dictionary with 5 × 2 × 3 × 2 = 60 total possible polygons, and hence 60 possible swarm behaviors.

B. Measuring Dictionary Sorting Proficiency

Although in theory SCINET is capable of controlling an arbitrary number of degrees of freedom (i.e., string characters), this scalability is limited in practice by the ability and ease by which the BCI operator can sort strings according to the swarm dictionary ordering. To be a useful approach, a typical human user must be able to quickly learn the swarm dictionary and subsequently sort any pair of strings, with high proficiency when the critical character is located at any position in the string. To evaluate these user capabilities in an isolated manner from the rest of the BCI system, we conducted a user study where participants (n = 150) used a point-and-click interface to select between configurations on a screen. Each participant was first presented with a set of graphical and text instructions explaining the polygon dictionary ordering and how to use it to sort a given string pair. Each participant was then presented with 150 randomly selected shape pairs from the dictionary (Fig. 2b), and asked to indicate which shape precedes the other in the dictionary ordering. We provided each participant with a visual aid to use as a reference during the task (Fig. 2a), which could be presented to a BCI operator in a practical setting.

C. Full System Evaluation

Beyond the interaction algorithm, there are a number of additional factors which can affect SCINET performance in the full system. Namely, the user must not only compare the current swarm configuration against their target string in the dictionary ordering, but must then issue a binary input via a mental command and subsequently observe the real-time...
changes in the swarm’s behavior. Due to practical effects such as user fatigue, the user’s error in issuing inputs may stray from the theoretical BSC assumed by the posterior matching algorithm. Since posterior matching assumes a fixed BSC crossover probability, it is unclear if non-ideal input statistics will result in poor system performance, and if such effects can be modeled. To evaluate SCINET in practice, we measure accuracy of a physical SCINET implementation against a simulation model that accounts for these practical effects.

As a pilot demonstration, the first author trained a synchronous EEG MI classifier using standard practices (Supplement B.3) and used the rules of posterior matching to control a simulated robot swarm (presented visually on a monitor, see Fig. 3b). In a series of repeat trials, target configurations were presented and MI commands were issued to steer the swarm towards the specified configuration (full system diagram in Supplement Fig. 6). In Section III-B we compare the resulting target convergence accuracy against that of a simulator replicating the empirically observed input error profile, to control for the effect of non-ideal input statistics. Specifically, we fit a piecewise cubic Hermite interpolating polynomial (PCHIP) to the observed empirical crossover probability (Fig. 3c) and used this profile to generate input errors in a posterior matching simulation. In each simulation trial, the correct posterior matching input at iteration $i$ was flipped with a BSC crossover probability given by the $i$th PCHIP value. Importantly, at each iteration the string posterior distribution was updated as if errors were generated according to a BSC with a constant crossover probability of 21.8%, which is the empirical input error averaged across all swarm control trials. This simulates the real-world discrepancy between the trained EEG classifier assuming a fixed BSC crossover probability for each trial (Supplement B.3) and the actual input error statistics varying across inputs. Posterior matching was simulated with this input error profile for 10,000 trials, where at the start of each trial a target configuration was selected uniformly at random from the dictionary (details in Supplement B.5).

The first author also demonstrated SCINET’s capability to be implemented in a (non-virtual) cyber-physical system by steering a physical robot swarm in two trials as a proof-of-principle to complement the virtual simulations of swarm behavior (Fig. 3a). The quantity of these trials was limited due to availability of the physical robot system and finite robot battery life. We implemented the same EEG training and experimental setup procedures as in the virtual swarm trials, with the only difference being that physical robots responded to user commands rather than virtual robots.

D. Generalizing Performance Tradeoffs in Simulation

Ultimately, the accuracy and number of controllable degrees of freedom (and hence the dictionary size) in SCINET is determined by the error rate of the input mechanism and budget...
Fig. 3. End-to-end testing of full system with EEG inputs and swarm control. SCINET was tested as a full system by the first author for controlling both physical (a) and virtual (b) robot swarms. (a) When controlling physical robots, the target configuration is presented to the user as an illuminated shape on the robot arena (accentuated here for visibility). (b) During virtual swarm control, the BCI user views a monitor that presents a target configuration (depicted as a shape outline) alongside the swarm’s current configuration. The virtual robots simulate realistic robot motion, and readjust their positions dynamically after each user input. (c) Non-stationary crossover probability versus number of inputs in virtual swarm control trials \((n = 70)\). At each input, we estimate the empirical crossover probability (blue, dotted line, with 95\% Wilson confidence interval [47]) as the fraction of trials where that input was decoded incorrectly with respect to the target configuration. A modified cubic function was fit to this changing crossover probability (orange, solid line) and used to generate errors in a realistic SCINET simulation. (d) Comparison of experimental and simulation configuration accuracy as a function of number of inputs until convergence, where the non-stationary crossover profile from (c) was used for simulating input errors. Results are binned into short, medium, and long trials by selecting bin edges at the 1/3 and 2/3 percentiles of virtual swarm convergence times, such that each bin includes approximately the same number of trials. Configuration accuracy within each bin is computed as the fraction of trials \((n = 10,000)\) converging successfully, where error bars depict the 95\% Wilson confidence interval. The overall configuration accuracy across all trials (regardless of number of inputs to converge) is also depicted. Experimental accuracy (binned and overall) closely matches that of the simulated model, suggesting that posterior matching (which assumes a fixed crossover probability) with input errors generated by the profile in (c) is an appropriate model for the observed experimental behavior. This suggests that the end-to-end system is performing as expected (once the input statistics are accounted for), and that it is reasonable to use this simulator to explore system behavior. See Supplement Secs. B.2–B.6 and D.2 for additional methods and discussion.

on the allowable number of inputs; increasing the controlled degrees of freedom requires additional inputs to refine effector behavior. To more fully explore this tradeoff, we use different input error profiles and dictionary sizes (corresponding to a variety of end-effector degrees of freedom) to simulate posterior matching as well as a baseline interaction algorithm (called stepwise search) that resembles discrete menu selection in existing BCIs (Supplement B.8). In stepwise search, each binary input updates the swarm’s guessed configuration by moving to the next string in the dictionary, in the direction indicated by the user’s input; the number of such steps needed for convergence scales linearly with the size of the dictionary.

In the data collected from a simple interface (with input characteristics reported in Fig. 3c), our proposed interaction mechanism can work well in some scenarios despite the relatively high overall error rate and the non-stationary error profile that nears chance probability (50\%) as the number of inputs increases. However, large dictionaries providing more resolution will suffer a performance bottleneck with this non-stationary error profile because they require more inputs for convergence. While developing high-performance input mechanisms is not the focus of this work, we evaluate SCINET system performance with realistic improved input mechanisms by simulating both posterior matching and stepwise search for a fixed crossover probability of 10\% (comparable to input errors seen in prior work [49]) and a variety of dictionary sizes (expressed as an equivalent number of degrees of freedom by subdividing each dictionary with an average alphabet size from our physical system).

1) Simulator Details: To generalize the performance of SCINET to arbitrary dictionaries, the simulator from Section II-C was modified slightly. To fully evaluate the tradeoff between achieved configuration accuracy and required number of inputs for each dictionary size, we disabled convergence for both posterior matching and stepwise search and instead output an instantaneous configuration estimate after

\[^1\]This algorithm is similar to the fixed offset [48] and sequential-select [38] policies explored in previous work on posterior matching-based BCIs.
Fig. 4. Performance as a function of number of inputs and dictionary size. We simulated the proposed posterior matching approach (solid lines) and stepwise search akin to discrete menu selection (dotted lines) over various dictionary sizes corresponding to different (estimated) end effector degrees of freedom. We evaluate both algorithms over two input error profiles: a fixed 10% crossover probability similar to prior reported decoding performance (a-c), and more adverse non-stationary errors generated according a model of our physical experiments (d-f). (a,d) Information transfer rate (ITR) [50], measuring the amount of information specified by a trial’s inputs with respect to dictionary size. Error bars are calculated as the ITR of the corresponding accuracy limits in (b) and (e). (b,e) Fraction of estimated configurations that are a perfect match with the target configuration, with 95% Wilson confidence intervals. (c,f) Absolute deviation between estimated and target configurations, where error bars depict 95% bootstrap confidence intervals over 10,000 samples (separate resampling for every number of inputs). By all metrics (a-f), posterior matching greatly outperforms stepwise search, with differences in performance becoming more drastic for larger dictionaries. Posterior matching obtains high configuration accuracies at high (>4) degrees of freedom with a modest number of inputs (additional discussion in Supplement D.3).
For this reason, we call the $b = 3$ case the “standard” degrees of freedom estimate, and $b = 5$ “conservative.”

To evaluate these simulation trials, we calculate information transfer rate (ITR) [50] from trial configuration accuracy, as follows: at $k$ inputs issued, let $P_k$ denote the error-free accuracy, which is calculated as the number of trials where the $k$th instantaneous estimate (i.e., $Z_{j^*}$) equals the ground truth target configuration, divided by the total number of simulation trials (1,000). ITR, denoted after $k$ inputs as $R_k$, is then calculated in units of bits as [50]

$$R_k = \log_2 N_d + P_k \log_2 P_k + (1 - P_k) \log_2 \frac{1 - P_k}{N_d - 1}.$$  

ITR represents the aggregate amount of information about the target configuration conveyed after $k$ inputs from the user to the swarm, and is a measure of configuration accuracy that also takes into account the increased difficulty of configuration selection with larger dictionary sizes. ITR can be also interpreted mathematically as the bit rate over a discrete memoryless channel where the target is selected with probability $P_k$, and any remaining configuration is erroneously selected with an equal probability of $\frac{1-P_k}{N_d-1}$. We also calculate an absolute deviation metric: let $Z_{j^*}(k)$ and $Z_i$ denote the unit interval representations (Supplement B.7) of the MAP estimate after $k$ inputs and the target configuration, respectively. Then absolute deviation is calculated as $|Z_{j^*}(k) - Z_i|$, and averaged over all trials for each simulation.

III. RESULTS

A. Sorting Proficiency User Study

Overall, participants were able to sort shape pairs with high accuracy. When evaluating sorting accuracy over all pairs of strings (Fig. 2c), most subjects sorted with nearly perfect accuracy (median 99.3% accuracy). Furthermore, response accuracy does not appear to decrease as the position of the critical character appears later in the string (median 100% accuracy for all characters, see Fig. 2d). When evaluating critical character performance for each individual participant, we also find that most participants exhibit non-decreasing or only modestly decreasing performance as character depth increases (Supplement Fig. 8 and D.1). Although a user’s capacity for learning and memorizing a dictionary ordering may create a performance bottleneck, this can be mitigated by providing users with a mnemonic aid to assist in recalling the ordering, as was done in our study. These results suggest that users can rapidly learn and apply string sorting in our heterogeneous dictionary, and that adding more characters (i.e., parameters) does not hinder a user’s ability to effectively compare pairs of strings across multiple parameters simultaneously.

B. Full System Experiments

1) Virtual Swarm Control: As one might anticipate, over the course of issuing a sequence of MI inputs, the error rate of user inputs (calculated with respect to the correct input according to the rules of posterior matching) varied as additional commands were issued (Fig. 3c). In theory, this error can be attributed to both the user error of issuing the incorrect posterior matching input, as well as classification error due to the MI detection algorithm classifying the input incorrectly. We conclude from the previous dictionary sorting user study that the former error source is small (estimated at 4.2%, see Fig. 2c), and therefore the increasing net input errors are likely due to degrading MI signal feature separation (Supplement Fig. 12). This effect is possibly due to user fatigue in issuing a sequence of inputs with minimal training, and resulted in an overall input error of 21.8%. Given previous work on MI inputs, this error rate is likely to be significantly improved with higher-fidelity interfaces and more extensive user training [49].

Despite nontrivial input error rates, this prototype system achieved an overall configuration selection accuracy of 75.7% (Fig. 3d), calculated as the fraction of trials where the swarm converged perfectly to the specified target with zero error; this greatly exceeds the accuracy of 1.67% that would be obtained by chance selection alone. Furthermore, we can account for these observed results with our simple simulation model on the non-stationary input statistics, which obtains a similar configuration accuracy (74.3%) to that observed in practice (Fig. 3d). Additionally, this model matches the observed behavior even when evaluating trials based on their required numbers of inputs to converge, which is a distinguishing element between trials since longer convergence is associated with increasing input errors and, therefore, with decreased performance.

2) Physical Swarm Control: The first author successfully used SCINET to steer the physical swarm to the correct target configuration across two trials (see supplement videos). Each physical swarm trial took approximately 10 minutes to complete, with the sources of significant latency being the time required for the swarm to converge to each estimated configuration, and the synchronous window required for EEG input commands (Supplement B.3). The swarm convergence latency is unavoidable, since robot speed is an inherent property of the swarm. Since the focus of this work is on developing an improved interaction algorithm rather than obtaining state of the art input detection, in classifying EEG inputs we opted to utilize a classic, straightforward technique rather than optimizing for EEG input latency. However, one could substitute the EEG input detection used here with any noninvasive binary input mechanism, such as asynchronous low-latency motor imagery detection [51]. Taken together, our results demonstrate that SCINET can achieve reasonable configuration accuracy despite the presence of non-stationary input errors, and that performance can be captured by a simple model. Additionally, the availability of a simulator that closely matches observed empirical behavior allows us to explore the performance of SCINET with more general dictionaries.

C. Generalized Simulation Experiments

To explore the relationship between number of issued inputs, dictionary size, and configuration accuracy, in Fig. 4a we plot the ITR (specified in bits per trial) of posterior matching and stepwise search for a 10% error input profile. The increasing ITR for posterior matching demonstrates that SCINET, with this simulated input mechanism, can achieve increasingly high information rates with larger dictionaries. The fact that ITR increases for posterior matching, both in terms of dictionary size and number of inputs, can be
interpreted to mean that even though larger dictionaries present more challenging configuration selection tasks, SCINET can accurately specify these configurations with enough inputs. The fraction of error-free configurations (i.e., perfectly achieving the desired configuration) also approaches 100% (Fig. 4b), even with large dictionaries and non-zero error rates in the user input. Finally, to study the rate of convergence of the estimated configuration to the target in the dictionary (which is not reflected in the fraction of error-free configurations), we also measure the absolute deviation of the estimated configuration from the target and observe that error decays quickly regardless of dictionary size (Fig. 4c).

In all metrics, posterior matching vastly outperforms discrete menu selection through a stepwise search approach. While larger dictionaries require more inputs to refine a configuration to a desired level of accuracy, SCINET with a fixed crossover probability still achieves high performance for large dictionaries in a modest number of inputs. We note that with a fixed input profile, SCINET can successfully control upwards of 6 separate degrees of freedom, which (to our knowledge) exceeds the current capabilities of non-invasive continuous control BCIs. We also plot the same performance metrics for simulations where input errors are generated according to the non-stationary profile observed in our physical experiments (Fig. 4d-f). Even with this adverse input characteristic, SCINET greatly outperforms stepwise search across all metrics. Although performance degrades for larger dictionary sizes, these larger dictionary sizes correspond to estimated degrees of freedom that lie beyond the control capabilities of typical noninvasive BCIs.

IV. DISCUSSION

The results in this work demonstrate how our proposed SCINET interaction algorithm significantly expands the capabilities of low-complexity BCIs to efficiently, robustly, and scalable control high-complexity effectors, while requiring no more than currently available signal acquisition hardware already in widespread development and use. The success of human users learning and sorting a heterogeneous shape dictionary supports the use of pairwise string sorting as a simple-to-use and tractable interface design that scales well with the complexity of the end effector system. When tested in a physical system, SCINET can perform well despite the presence of non-stationary input errors, validating the deployment of posterior matching control over a heterogeneous dictionary in a practical setting. By extending our experimental results to a range of dictionary sizes and input mechanism fidelities through realistic simulations, we find that posterior matching outperforms a baseline algorithm comparable to discrete menu selection and can control a large number of estimated degrees of freedom with only a modest number of inputs.

Although only the first author conducted a demonstration of the full cyber-physical SCINET system (i.e., EEG commands issued to control virtual or physical robots), these trials were conducted in a repeated and controlled fashion ($n = 70$) on a realistic swarm simulator and therefore provide initial evidence for the real-world potential of our system. It is also important to note that the practice of detecting motor imagery inputs from EEG is well-established, and therefore our evaluation instead focuses on the ability of human users to navigate our heterogeneous dictionary, and on the convergence of posterior matching within this dictionary over various input error profiles. We have shown through a large number of repeat user interface trials ($n = 150$ subjects) that humans can successfully and scalably perform dictionary navigation via paired comparisons, and we have also validated the use of posterior matching in this setting through extensive simulations.

While posterior matching with a heterogeneous dictionary was implemented here for the control of robot swarms, the general technique is applicable to any setting where each effector parameter can be assigned its own ordered alphabet. Importantly, our approach has the flexibility for a system designer to select a dictionary size based on their effector’s behavioral specifications such as allowable number of user inputs, minimum configuration accuracy, and number of effector parameters (i.e., degrees of freedom). Once the designer decides on a fixed number of dictionary elements, they can then distribute this fixed number of elements among their degrees of freedom in a customized manner by tuning the size of each character’s alphabet, allowing for variable resolutions between parameters. As we demonstrate through extensive simulations, these tradeoffs between controllable degrees of freedom and number of required inputs can be adapted to the error profile of the input mechanism. This is in contrast to invasive neural measurement systems, which in general do not allow for tradeoffs between input accuracy, number of inputs, and controllable degrees of freedom. Instead, such invasive methods utilize a single high-fidelity measurement to instantaneously extract a total system state from the BCI user, while imposing a significant burden and disruption on the user by requiring a surgical procedure. By iteratively refining effector behavior through a sequence of low-complexity inputs rather than imposing such stringent requirements on the input mechanism, SCINET complements years of research devoted to improving the input mechanisms of BCIs by instead fundamentally redesigning how inputs are utilized.

REFERENCES

[1] S. R. Soekadar et al., “Hybrid EEG/EOG-based brain/neural hand exoskeleton restores fully independent daily living activities after quadriplegia,” Sci. Robot., vol. 1, no. 1, Dec. 2016.
[2] J. L. Collinger et al., “High-performance neuroprosthetic control by an individual with tetraplegia,” Lancet, vol. 381, no. 9866, pp. 557–564, Feb. 2013.
[3] L. R. Hochberg et al., “Reach and grasp by people with tetraplegia using a neurally controlled robotic arm,” Nature, vol. 485, no. 7398, pp. 372–375, May 2012.
[4] L. R. Hochberg et al., “Neuronal ensemble control of prosthetic devices by a human with tetraplegia,” Nature, vol. 442, no. 7099, pp. 164–171, Jul. 2006.
[5] B. Woodinger, J. E. Downey, E. C. Tyler-Kabara, A. B. Schwartz, M. L. Boninger, and J. L. Collinger, “Ten-dimensional anthropomorphic arm control in a human brain–machine interface: Difficulties, solutions, and limitations,” J. Neural Eng., vol. 12, no. 1, Dec. 2014, Art. no. 016011.
[6] R. Leeb, D. Friedman, G. R. Müller-Putz, R. Scherer, M. Slater, and G. Pfurtscheller, “Self-paced (asynchronous) BCI control of a wheelchair in virtual environments: A case study with a tetraplegic,” Comput. intell. Neurosci., vol. 2007, pp. 79642–79648, Jan. 2007.
[17] B. Rebsamen et al., “Controlling a wheelchair indoors using thought,” IEEE Intell. Syst., vol. 22, no. 2, pp. 18–24, Mar. 2007.

[18] D. Huang, K. Qian, D. Fei, W. Jia, X. Chen, and O. Bai, “Electroencephalography (EEG)-based brain–computer interface (BCI): A 2-D virtual wheelchair control based on event-related desynchronization/synchronization and state control,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 20, no. 3, pp. 379–388, May 2012.

[19] Y. Li, J. Pan, F. Wang, and Z. Yu, “A hybrid BCI system combining P300-based-SVEP and control to wheelchair,” IEEE Trans. Biomed. Eng., vol. 60, no. 11, pp. 3156–3166, Nov. 2013.

[20] J. Long, Y. Li, H. Wang, T. Yu, J. Pan, and F. Li, “A hybrid brain computer interface to control the direction and speed of a simulated or real wheelchair,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 20, no. 5, pp. 720–729, Sep. 2012.

[21] J. Li, J. Liang, Q. Zhao, J. Li, K. Hong, and L. Zhang, “Design of a noninvasive wheelchair system directly steered by human thoughts,” Int. J. Neural Syst., vol. 23, no. 3, 2013, Art. no. 1350013.

[22] T. Carlson and J. Del R Millan, “Brain-controlled wheelchairs: A robotic architecture,” IEEE Robot. Autom. Mag., vol. 20, no. 1, pp. 65–73, Mar. 2013.

[23] R. Zhang et al., “Control of a wheelchair in an indoor environment based on a brain–computer interface and automated navigation,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 18, no. 6, pp. 590–598, Dec. 2010.

[24] B. Blankertz, G. Dornhege, M. Krauledat, K.-R. Müller, and G. Curio, “The non-invasive Berlin brain–computer interface: Fast acquisition of effective performance in untrained subjects,” NeuroImage, vol. 37, no. 2, pp. 539–550, 2007.

[25] J. R. Wolpaw and D. J. McFarland, “Control of a two-dimensional movement signal by a noninvasive brain–computer interface in humans,” Proc. Nat. Acad. Sci. USA, vol. 101, no. 51, pp. 17849–17854, Dec. 2004.

[26] B. Xia, D. An, C. Chen, H. Xie, and J. Li, “A mental switch-based asynchronous brain–computer interface for 2D cursor control,” in Proc. 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC), Jul. 2013, pp. 3101–3104.

[27] D. J. McFarland, W. A. Sarnacki, and J. R. Wolpaw, “Electroencephalographic (EEG) control of three-dimensional movement,” J. Neural Eng., vol. 7, no. 3, 2010, Art. no. 036007.

[28] J. Long, Y. Li, T. Yu, and Z. Gu, “Target selection with hybrid feature for 3D-based 2D cursor control,” IEEE Trans. Biomed. Eng., vol. 59, no. 1, pp. 132–140, Jan. 2012.

[29] L. J. Trejo, R. Rosipal, and B. Matthews, “Brain–computer interfaces for 1-D and 2-D cursor control: Designs using volitional control of the EEG spectrum or steady-state visual evoked potentials,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 14, no. 2, pp. 225–229, Jun. 2006.

[30] (2020). Imagining a New Interface: Hands-Free Communication Without Saying a Word. [Online]. Available: https://tech.fb.com/imaging-a-new-interface-hands-free-communication-without-saying-a-word.

[31] E. Musk, “An integrated brain-machine interface platform with thousands of channels,” J. Med. Internet Res., vol. 21, no. 10, Oct. 2019. Art. no. e16194. [Online]. Available: http://www.jmir.org/2019/10/e16194/.

[32] (2018). Nonsurgical Neural Interfaces Could Significantly Expand Use of Neurotechnology. [Online]. Available: https://www.darpa.mil/news-events/2018-03-16.

[33] E. J. Pratt et al., “Kernel Flux: A whole-head 432-magnetometer optically-pumped magnetoencephalography (0-MEG) system for brain activity imaging during natural human experiences,” in Proc. SPIE, vol. 11700, 2021, pp. 162–179.

[34] R. Scherer, G. R. Müller, C. Neuper, B. Graimann, and G. Pfurtscheller, “A neurotechnology-controlled EEG-based virtual keyboard: Improvement of the spelling rate,” IEEE Trans. Biomed. Eng., vol. 51, no. 6, pp. 979–984, Jun. 2004.