

# Linear Diophantine equation (LDE) decoder: A training-free decoding algorithm for multifrequency SSVEP with reduced computation cost

Jing Mu<sup>1</sup>  | Ying Tan<sup>1,2</sup>  | David B. Grayden<sup>2,3</sup>  | Denny Oetomo<sup>1,2</sup> 

<sup>1</sup>Department of Mechanical Engineering, The University of Melbourne, Parkville, Victoria, Australia

<sup>2</sup>Graeme Clark Institute, The University of Melbourne, Parkville, Victoria, Australia

<sup>3</sup>Department of Biomedical Engineering, The University of Melbourne, Parkville, Victoria, Australia

## Correspondence

Jing Mu, Department of Mechanical Engineering, The University of Melbourne, Grattan Street, Parkville, Victoria 3010, Australia.  
Email: j.mu@student.unimelb.edu.au

## Funding information

Valma Angliss Trust

## Abstract

Multifrequency steady-state visual evoked potentials (SSVEPs) have been developed to extend the capability of SSVEP-based brain-machine interfaces (BMIs) to complex applications that have large numbers of targets. Even though various multifrequency stimulation methods have been introduced, the decoding algorithms for multifrequency SSVEP are still in early development. The recently developed multifrequency canonical correlation analysis (MFCCA) was shown to be a feasible training-free option to use in decoding multifrequency SSVEPs. However, the time complexity of MFCCA is shown to be  $O(n^3)$ , which will lead to long computation time as  $n$  grows, where  $n$  represents the input size in decoding. In this paper, a novel decoding algorithm is proposed with the aim to reduce the time complexity. This algorithm is based on linear Diophantine equation solvers and has a reduced computation cost  $O(n \log n)$  while remaining training-free. Our simulation results demonstrated that linear Diophantine equation (LDE) decoder run time is only one fifth of MFCCA run time under respective optimal settings on 5-s single-channel data. This reduced computation cost makes it easier to implement multifrequency SSVEP in real-time systems. The effectiveness of this new decoding algorithm is validated with nine healthy participants when using dry electrode scalp electroencephalography (EEG).

## KEYWORDS

brain-computer interface (BCI), brain-machine interface (BMI), decoder, linear Diophantine equation, multifrequency, steady-state visual evoked potential (SSVEP)

## 1 | INTRODUCTION

Steady-state visual evoked potentials (SSVEPs) are natural responses of the brain that react to periodically flicking visual stimulation [1, 2]. They are considered a robust modality of brain activities that could be captured noninvasively through, for example, electroencephalography (EEG). SSVEP is widely used in noninvasive brain-machine interfaces (BMIs) because of its relatively

high signal-to-noise ratio, minimal training requirements, and capability of facilitating larger number of commands at once compared to other brain activity modalities such as motor imagery [3].

In SSVEP-based BMIs, there are usually three major components: The visual stimulation medium that produces visual stimulation to the user, the brain signal capturing system that records the evoked SSVEP from the user, and the decoding algorithm (decoder) that deciphers

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2023 The Authors. Asian Journal of Control published by John Wiley & Sons Australia, Ltd on behalf of Chinese Automatic Control Society.

user intentions. Based on the knowledge that SSVEP is frequency-locked to the frequency of the modulation of the visual stimulation, the interface is usually set up to have multiple targets in the scene, each labeled with a unique frequency that will be delivered through flickering. The targets could be light emitting diodes (LEDs) placed on or near objects to represent potential actions, items, or coordinates of reaching [4–7] or represented on a computer screen with each target block representing a character in a BMI speller or commands for controlling the computer or other devices [8–10]. In order to identify the user's intended target from all targets presented in the interface, the decoding algorithm analyses the frequency components of the collected brain signals that contain SSVEP and make a decision based on the dominate frequency features. In a typical SSVEP setup, the evoked SSVEP contains the stimulation frequency  $f$ , as well as the harmonics of this frequency  $2f, 3f, \dots$  [1, 11].

One of the limitations in traditional SSVEP-based BMIs is that the number of targets is constrained by the limited responsive range of SSVEP [1] and the existence of harmonics, which may lead to misclassification if a frequency and its harmonics are used together in the interface at the same time. This slows down the development of BMI in terms of increasing the command processing capacity (number of commands) [12]. To address this problem, multifrequency SSVEP stimulation methods have been introduced aiming to increase the number of targets presentable with limited frequencies available [13–17]. However, the decoders for multifrequency SSVEP have not been widely explored yet. Existing multifrequency SSVEP decoders include power spectral density based analysis (PSDA) [15, 17], multifrequency canonical correlation analysis (MFCCA) [18], and training-based algorithms for each individual user or use case [13, 19]. Training-based algorithms produce higher classification accuracies compared to the two training-free methods but require additional efforts in training for each user and interface setup. PSDA and MFCCA support plug-and-play that improves practicality of the BMI. However, PSDA usually has limited decoding accuracy as it does not fully consider the complex frequency features in multifrequency SSVEP, which contain not only the stimulation frequencies and their harmonics (as in single-frequency SSVEP) but also the linear interactions between the stimulation frequencies [16]. MFCCA has shown its advantage in multifrequency SSVEP decoding by introducing the linear interactions in decoding without the need of training [18], but one major problem of MFCCA is that it was developed based on canonical correlation analysis (CCA) [20], which has a high time complexity. The asymptotic time complexity of CCA is  $O(lD^2) + O(D^3)$  (bounded by  $O(n^3)$ , where  $n$  represents the input size in decoding), where  $l$

is the number of samples in the SSVEP recording,  $D = \max(D_{\text{data}}, D_{\text{ref}})$ ,  $D_{\text{data}}$  is the number of channels in SSVEP recording,  $D_{\text{ref}}$  is the number of rows in the reference set in CCA formulation, which grows as the frequency feature becomes more complex [21]. MFCCA uses the same operations as CCA, but a more complex reference set (larger  $D_{\text{ref}}$ ). So, MFCCA may end up requiring long processing time, which may limit the ability of MFCCA to be applied to problems with high complexity or in real-time systems. In single-frequency SSVEP, the decoding problem has also been approached by treating it as a blind source separation problem, where principal component analysis (PCA) and independent component analysis (ICA) are widely used, and decoding algorithms such as minimum energy combination (MEC) [22] based on PCA and local Fourier-ICA [23] based on ICA have been developed. However, multifrequency SSVEP decoding is a more complex problem to solve as it contains nonlinear interactions of multiple input frequencies in the recorded EEG. As the existing techniques focus on linear maps, they are not directly applicable to multifrequency SSVEP. Hence, there is a need to develop novel generalized decoding algorithms for multifrequency SSVEP, not only to lower the computational cost but also to provide alternatives in multifrequency SSVEP decoding and improve decoding accuracy where possible.

It was understood that the observed interactions between input frequencies can be written as integer linear combinations of the input frequencies [16, 18]. This relationship with integer coefficients fits into a linear Diophantine equation (LDE) problem. A LDE is a family of linear polynomial equations that has integer coefficients and integer unknowns. The simplest LDE can be written as

$$\alpha x + \beta y = \gamma, \quad (1)$$

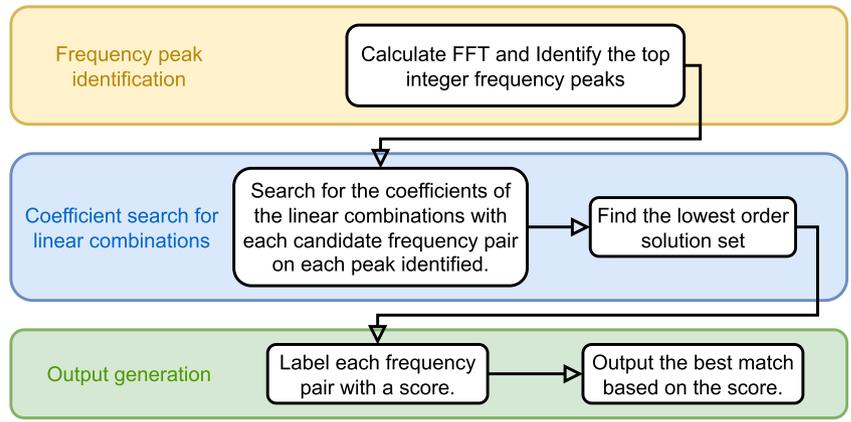
where  $\alpha, \beta$ , and  $\gamma$  are integers and  $x, y \in \mathbb{Z}$  are unknown integers to be estimated. Here,  $\mathbb{Z}$  is the set containing all integers. In dual-frequency SSVEP, the integer linear interactions we have observed can be written as

$$f_p = c_1 f_1 + c_2 f_2, \quad (2)$$

where  $f_p$  is a peak frequency detected,  $c_1$  and  $c_2$  are unknown integer coefficients, and  $f_1$  and  $f_2$  are the two input frequencies. When  $f_1$  and  $f_2$  are both integers, the frequency interactions in multifrequency SSVEP become LDEs.

This work explores the possibility of decoding multifrequency SSVEP by formulating the problem into solving a set of LDEs. This decoding algorithm will be hereafter called the LDE decoder. The effectiveness of LDE will be assessed and compared to MFCCA with EEG data collected from nine participants. For simplicity in explana-

**FIGURE 1** Framework of the algorithm.  
[Color figure can be viewed at wileyonlinelibrary.com]



tion, dual-frequency SSVEP is used to illustrate the idea in this work; however, the method is also applicable to multifrequency scenarios. The main contribution of this work lies in the formulation of the multifrequency SSVEP decoding problem, which allows the application of the off-the-shelf LDE solver. This formulation along with the existing solution reduces the computation cost in multifrequency SSVEP decoding, enabling the multifrequency SSVEP to be implemented in real-time systems.

## 2 | OVERVIEW OF THE LDE DECODER

### 2.1 | Assumptions

In a multifrequency SSVEP decoding problem, the standard assumptions used are summarized as follows [18]:

**Assumption 1.** The frequency components in the resulting SSVEP response contain the stimulation frequencies and their harmonics as well as frequencies that are linear combinations of the input frequencies with integer coefficients.

**Assumption 2.** Frequencies with lower order in the linear combinations are more likely to be observed in the evoked SSVEP.

In order to formulate multifrequency interactions into LDEs, the following assumptions are needed:

**Assumption 3.** The frequencies used in stimulation are all integers.<sup>1</sup>

**Assumption 4.** The algorithm has the knowledge of a candidate set of stimulation frequency combinations.

While the algorithm requires knowledge of a candidate set of stimulation frequencies, it does not impose any constraints in the selection of frequencies.

### 2.2 | Framework

The developed decoding algorithm has three major parts: Frequency peak identification, coefficient search for linear combinations, and output generation. Figure 1 depicts the general structure of the algorithm. Each block will be explained in detail in Section 3.

### 2.3 | Problem formulation for coefficient search

Assume:

1.  $n$  integer peaks  $\mathbf{P} = [p_1, p_2, \dots, p_n]^T$  were identified from frequency peak identification,
2. the peak frequencies are linear combinations of the two input stimulation frequencies,
3. the candidate set of frequency pairs is known,

$$\mathbf{F}_{\text{cand}} = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \\ \vdots & \vdots \\ f_{n1} & f_{n2} \end{bmatrix}, \quad (3)$$

where  $f_{ij}, i = 1, 2, \dots, n, j = 1, 2$  is the frequency  $f_j$  in the  $i^{\text{th}}$  pair of frequency candidates.

Then, the problem can be formulated as

$$\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ \vdots & \vdots \\ c_{n1} & c_{n2} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}, \quad (4)$$

$$\mathbf{C} \quad \mathbf{F}^T = \quad \mathbf{P},$$

where  $f_1$  and  $f_2$  are two unknown integer frequencies in the given set,  $p_i, i = 1, 2, \dots, n$  is the  $i^{\text{th}}$  highest integer peak on FFT,  $c_{ij}, i = 1, 2, \dots, n, j = 1, 2$  is unknown

<sup>1</sup>This assumption can be relaxed. See Section 3.5.2 for details.

integer coefficient for  $f_j$  that contributes to peak  $p_i$ , and  $\mathbf{C}$  is the matrix that contains all unknown integer coefficients.

In this problem,  $p_i$  is known and is a positive integer,  $c_{ij}$  is an unknown integer, and  $f_1$  and  $f_2$  are the two positive integers we are solving for. In other words, in the formulation above (Equation 4), both  $\mathbf{C}$  and  $\mathbf{F}$  are unknowns. There are  $2n + 2$  unknowns but only  $n$  constraints, which makes the problem difficult to solve. However, we know the candidate set of  $f_1$  and  $f_2$ . Therefore, this problem can be reduced to repeatedly solve for the following problem.

For each  $f_1$  and  $f_2$  combination, for each  $p$  in  $\mathbf{P}$ , solve for  $c_1$  and  $c_2$ ,

$$\begin{bmatrix} f_1 & f_2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = p, \quad (5)$$

$$\mathbf{F} \quad \mathbf{C}^* = p,$$

where  $f_1$ ,  $f_2$ , and  $p$  are given by the known candidate frequency set and identified peak, respectively.

We rewrite Equation (5) into a general form

$$\alpha x + \beta y = \gamma, \quad (6)$$

where  $\alpha$ ,  $\beta$ , and  $\gamma$  are integers and  $x$  and  $y$  are unknown integers to be sought. Since  $\alpha$ ,  $\beta$ , and  $\gamma$  are known integers, this is now in a standard LDE form, which enables us to make use of existing algorithms to solve it, such as the extended Euclidean algorithm [24]. This work makes use of the existing algorithms to fit into our application—SSVEP decoding.

### 3 | LDE DECODING ALGORITHM

In the framework (Figure 1) introduced above, there are three major steps in the proposed LDE decoding algorithm, which are elaborated below. A solution to the LDE was derived by Gilbert and Pathria using unimodular row reduction [24]. The LDE decoder developed here is based on the algorithm of Gilbert and Pathria [24].

**Algorithm 1** Solve linear Diophantine equation with extended Euclidean algorithm

**Solve:**  $\mathbf{F}\mathbf{C}^* = p$ , where  $\mathbf{F} = [f_1 \ f_2]$ ,  $\mathbf{C}^* = [c_1 \ c_2]^T$ , as defined in Eq. (5).

**Input:** Candidates of  $\mathbf{F}$ , list of  $p$ .

```

1: for each candidate of  $\mathbf{F}$  ( $f_1 \ f_2$  combination) do
2:    $d \leftarrow \text{gcd}(f_1, f_2)$  ▷ Find the greatest common divisor (gcd)  $d$  of  $f_1$  and  $f_2$ 
3:   for each  $p$  do
4:     if  $p$  is divisible by  $d$  then
5:        $m \leftarrow p/d$ 
6:     else
7:       break; ▷ There is no integer solution with the current  $\mathbf{F}$  and  $p$ . Move to the next candidate of  $\mathbf{F}$ 
8:     end if
9:      $\mathbf{M} \leftarrow [\mathbf{F}^T | \mathbf{I}]$  ▷ Here,  $\mathbf{F}$  is of size  $1 \times 2$ ,  $\mathbf{I}$  is an identity matrix in  $R^{2 \times 2}$ 
From here, we perform unimodular row reduction until  $\mathbf{M} = [\mathbf{R} | \mathbf{T}]$ , where  $\mathbf{R} = [d \ 0]^T$  with  $d$  being the greatest common divisor, and  $\mathbf{T}$  is the resulting  $2 \times 2$  matrix from the unimodular row operation
10:    do ▷ Unimodular row reduction
11:       $i \leftarrow$  index of the entry with smaller absolute value in  $\mathbf{M}(:, 1)$  ▷  $\mathbf{M}(:, 1)$ : first column in  $\mathbf{M}$ 
12:       $j \leftarrow$  index of the entry with larger absolute value in  $\mathbf{M}(:, 1)$ 
13:       $n \leftarrow \text{floor}(\mathbf{M}(j, 1)/\mathbf{M}(i, 1))$  ▷  $\text{floor}$ : an operation that rounds real number to the nearest integer that is not larger than the current value
14:       $r \leftarrow \text{mod}(\mathbf{M}(j, 1), \mathbf{M}(i, 1))$  ▷  $\text{mod}(\text{num}, \text{den})$ : an operation that gets the remainder of a division  $\frac{\text{num}}{\text{den}}$ 
15:      Subtract  $n$  times row  $i$  from row  $j$ .
16:      while  $\mathbf{M}(i, 1) \neq 0 \ \&\& \ \mathbf{M}(j, 1) \neq 0$  ▷ Continue if neither of the elements is zero
17:        if  $\mathbf{M}(1, 1) == 0$  then
18:          Interchange the two rows. ▷ Rearrange the matrix to have the zero on the last row
19:        end if ▷ Now we have  $\mathbf{M} = [\mathbf{R} | \mathbf{T}]$ , where  $\mathbf{R} = [d \ 0]^T$ 
20:         $c_1 \leftarrow \mathbf{T}(1, 1)m + \mathbf{T}(2, 1)k$ ,  $c_2 \leftarrow \mathbf{T}(1, 2)m + \mathbf{T}(2, 2)k$ ,  $k \in \mathbb{Z}$  ▷  $c_1$  and  $c_2$  can be written as functions of  $k$ . We will find the suitable  $k$  in Algorithm 2
21:    end for
22:  end for

```

**Output:** Expressions of  $c_1$  and  $c_2$ .

### 3.1 | Frequency peak identification

In the frequency peak identification step, after any pre-processing of the recorded data, the fast Fourier transform (FFT) was first performed. When dealing with a short period of EEG recording, data could be zero-padded to improve frequency resolution. Once the FFT is obtained, we could then read the top  $n$  integer frequency peaks  $\mathbf{P} = [p_1, p_2, \dots, p_n]^T$  with a small tolerance  $\epsilon$ , for example,  $\epsilon = 0.1$  Hz. This  $\epsilon$  can be tuned to suit the requirements or experimental setup in different studies.

### 3.2 | Coefficient search for linear combinations

As mentioned in Section 2.1, it was assumed that the LDE decoder is based upon the candidate set of frequency pairs. In this step, we need to check if integer coefficients can be found in a given range for each pair of frequency candidates to generate the identified peak frequencies from the previous step.

---

**Algorithm 2** Find  $k$  that minimizes  $|c_1| + |c_2|$

---

**Solve:**  $\min_{k \in \mathbb{Z}} |c_1| + |c_2|$ , where  $c_1 = a_1k + b_1$ ,  $c_2 = a_2k + b_2$ .

---

**Input:** Expressions of  $c_1$  and  $c_2$ .

---

- 1:  $i \leftarrow$  index of the larger in  $|a_1|$  and  $|a_2|$
  - 2:  $k' \leftarrow -b_i/a_i$  ▷ Solve  $a_i k' + b_i = 0$
  - 3:  $k_1 \leftarrow \text{floor}(k')$ ,  $k_2 \leftarrow \text{ceil}(k')$  ▷ Candidates of integer solutions
  - 4: **for**  $j=1,2$  **do**
  - 5:      $\text{order}_j \leftarrow |a_1k_j + b_1| + |a_2k_j + b_2|$  ▷ Calculate the order with each candidate  $k$
  - 6: **end for**
  - 7:  $l \leftarrow$  index of the smaller of  $\text{order}_1$  and  $\text{order}_2$  ▷ Find the  $k$  that result in the smallest  $|c_1| + |c_2|$
  - 8:  $k^* \leftarrow k_l$
  - 9:  $\text{order} \leftarrow \text{order}_l$
- 

**Output:**  $k^*$  and  $\text{order}$ .

---

#### 3.2.1 | Solving the LDE for the coefficients

The problem formulated in Section 2.3 can be solved with extended Euclidean algorithm, which is based on the idea of unimodular row reduction [24]. The algorithm is written in Algorithm 1. Note that there are infinite sets of solutions to a LDE of the format  $\alpha x + \beta y = \gamma$ . The results are written as functions of  $k$  ( $k \in \mathbb{Z}$ ).

Algorithm 1 starts with identifying if integer solution exists with the current candidate  $\mathbf{F}$  and peak  $p$  defined in Equation (5) by calculating the greatest common divisor  $d$  and checking if  $p$  is divisible by  $d$ . If yes, then we continue solving for the solutions; if not, we label it as “no solution” and move to the next candidate  $\mathbf{F}$ . In a case where there exists an integer solution, unimodular row reduction is performed to derive expressions of the coefficients.

#### 3.2.2 | Finding the lowest order solution

Based on our observations from pilot experiments, linear combinations of the two frequencies with large absolute values on the coefficients are often attenuated due to the nature of SSVEP and hardware limitations. In general, the larger the coefficients are, the weaker they show on the FFT plot. Therefore, not all solutions from Algorithm 1 are feasible solutions. For example, we can generate any integer number with integer linear combination of two prime numbers; however, the coefficients could be very large with some selection of the integer and the two prime numbers. Here, a bound was introduced to the sum of the absolute value of the two coefficients, or we could call it the order of the linear combination,  $\text{order} = |c_1| + |c_2| \in [\underline{c}, \bar{c}]$ . In most cases,  $\underline{c} = 1$  so that fundamental frequencies are included in the calculation. Hence, we need to find the  $k$  value that corresponds to the minimum order if there exists a solution to the LDE.

The optimal  $k = k^*$  corresponding to the lowest order coefficients can be found by solving

$$\operatorname{argmin}_{k \in \mathbb{Z}} |c_1| + |c_2|, \quad (7)$$

where  $c_1 = a_1k + b_1$  and  $c_2 = a_2k + b_2$ ,  $k \in \mathbb{Z}$ . Algorithm 2 is used to solve this optimization problem.

If  $\text{order} \in [\underline{c}, \bar{c}]$ , then we have a valid solution; otherwise, we will discard this solution and label it as “no solution.”

#### 3.2.3 | Proof of the optimality of $k^*$

Let

$$c_1(k) = a_1k + b_1,$$

$$c_2(k) = a_2k + b_2,$$

where  $a_1$ ,  $b_1$ ,  $a_2$ , and  $b_2$  are given integers.

Define a cost function  $c(k) = |c_1(k)| + |c_2(k)|$  with the integer variable  $k$ .

$$\text{Let } k_1 = \frac{b_1}{a_1} \text{ and } k_2 = \frac{b_2}{a_2}.$$

**Lemma 1.** *The optimization problem*

$$\min_{k \in \mathbb{Z}} c(k)$$

has the solution

$$c(k^*) = \min_{k \in \mathbb{S}} c(k),$$

where  $\mathbb{S} = \{\lfloor k' \rfloor, \lceil k' \rceil\}$ ,  $k' = k_i$ ,  $i = \arg \max_{i \in \{1,2\}} a_i$ .

*Proof.* Lemma 1 requires a minimization on sum of absolute functions. Hence, the objective function needs to be analyzed piece-wise. Let  $k'$  be the optimal solution in  $k \in \mathbb{R}$  and  $k^*$  be the optimal solution in  $k \in \mathbb{Z}$ .

fore,  $c(k)$  must also be a linear function of  $k$  in  $[k_2, k_1]$ . Let  $c_1(k_2) = A$ ,  $c_2(k_1) = B$ , then the line that represents  $c$  has to go through  $(k_2, A)$  and  $(k_1, B)$ . Hence, the minimum value locates at  $\min_k(A, B)$ , in other words, either  $k_1$  or  $k_2$ . Since  $k_1$  and  $k_2$  are the roots of the two functions, the comparison of  $c$  at  $k_1$  and  $k_2$  can be converted to the comparison of the absolute value of slopes of the two functions ( $|a_1|$  and  $|a_2|$ ). The larger the absolute value of slope is, the faster it grows as  $k$  moves away from its root and gives a larger  $c$ . Hence, the minimum value falls on the root of the function with larger absolute value of slope.

Figure 2 is a graphical demonstration of the optimality of  $k'$ . In this example,  $|a_1| < |a_2|$  and, therefore,  $k' = k_2$ .

Now we have obtained the optimal solution  $k'$ ,  $k' \in \mathbb{R}$ . Because the function  $c(k)$  is convex, the optimal

---

### Algorithm 3 Produce decoder output

---

```

1: for each  $f_1, f_2$  combination do
2:   Initialize: int counter = 0;
3:   int totalOrder = 0;
4:   for each  $p$  do
5:     if there exists a valid solution of  $c_1$  and  $c_2$  such that  $|c_1| + |c_2| \in [\underline{c}, \bar{c}]$  then
6:       counter+ = 1;
7:       totalOrder+ = order;
8:     end if
9:   end for
10:  NumOfValidSol(i) ← counter,  $i = 1, 2, \dots, m$ 
11:  Order(i) ← totalOrder,  $i = 1, 2, \dots, m$ 
12: end for
13: find the maximum value in NumOfValidSol and its index  $n$ 
14: if there exists multiple entries with the maximum value then
15:   find the one with minimum value in Order and its index  $n$ 
16: end if

```

▷ order is one of the outputs in Algorithm 2

▷  $m$  is the total number of  $f_1, f_2$  candidates  
 $i$  is the index of the current  $f_1, f_2$  pair

---

**Output:** the  $n^{\text{th}}$   $f_1, f_2$  candidates.

---

For simplicity, assume  $k_1 > k_2$ .

- i. In  $(-\infty, k_2]$ ,  $c$  increases as  $k$  decreases, because the slopes of both functions are negative in this region. Hence, the minimum value can be easily found at  $k' = k_2$ .
- ii. In  $[k_1, -\infty)$ ,  $c$  increases as  $k$  increases, because the slopes of both functions are positive in this region. Hence, the minimum value can be easily found at  $k' = k_1$ .
- iii. In  $[k_2, k_1]$ , as we know that  $c_1$  and  $c_2$  are linear functions of  $k$  and  $c(k) = |c_1(k)| + |c_2(k)|$ . There-

solution  $k^*, k^* \in \mathbb{Z}$  must be one of the integer values adjacent to  $k'$ . Hence,  $k^* = \operatorname{argmin}_{k \in \{\lfloor k' \rfloor, \lceil k' \rceil\}} c(k)$ . □

### 3.3 | Output generation

With the algorithms explained above, we can now check if each peak  $p$  allows us to find a set of coefficients  $c_1$  and  $c_2$  for the current  $f_1, f_2$  combination that falls within a range  $|c_1| + |c_2| \in [\underline{c}, \bar{c}]$ . In cases where a solution exists but the coefficients do not fall within the given range, this solution is considered invalid.

The output can be produced by counting the number of valid solutions we can obtain under each frequency

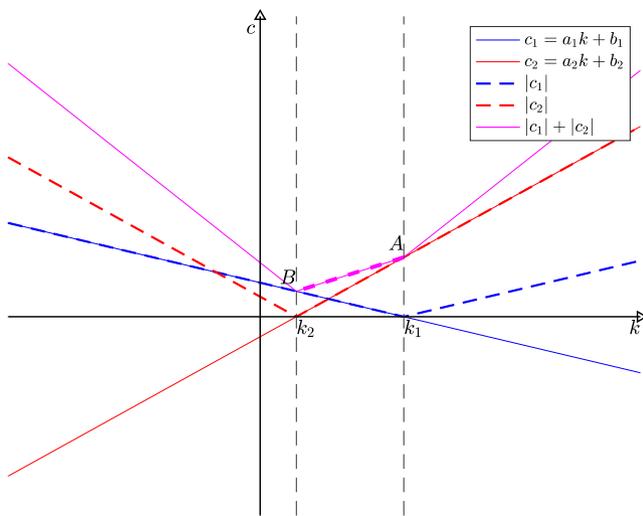


FIGURE 2 Graphical demonstration of the optimality of  $k'$ . [Color figure can be viewed at wileyonlinelibrary.com]

combination (label each frequency pair with a score) and choosing the combination that has the highest number of valid solutions as the decoder output (output the best match based on the score). This is explained in Algorithm 3. Here, a second source of information—total order from all valid solutions—was used to deal with cases where multiple frequency pairs yield the same number of valid solutions. In such cases, the pair with the least total order is regarded as the best match and returns by the algorithm.

### 3.4 | Time complexity of LDE

Based on the algorithm above, the computation pipeline of LDE can be summarized with asymptotic time complexity of each step labeled in parentheses:

- calculate FFT ( $O(n \log n)$ ),
- identify top peaks from FFT ( $O(n \log n)$ ),
- search for integer solution for each peak in each candidate pair ( $O(n)$ ) [25],
- find maximum value in the list for output generation ( $O(n)$ ).

All the steps above can be bounded by asymptotic time complexity  $O(n \log n)$ . Therefore, the overall asymptotic time complexity of LDE is  $O(n \log n)$ .

### 3.5 | Extend LDE decoders to more general cases

So far, LDE has been introduced and a detailed explanation given of how the algorithm works in decoding multifrequency SSVEP under the assumptions stated in Section 2.1. This section explains how LDE could be applied to a special case—single-frequency SSVEP decoding—and how Assumption 3 could be relaxed to

make LDE more powerful in multifrequency SSVEP decoding.

#### 3.5.1 | Special case: Decode single-frequency SSVEP

The proposed LDE decoder can be easily adjusted to work with single-frequency SSVEP. This can be done by replacing the second and subsequent frequency inputs to 0. By doing this, the algorithm is then simplified to a harmonics search problem instead of a coefficient search as explained in Section 3.2. Therefore, the more harmonics that can be identified in the top peaks, the more likely this candidate frequency is the stimulation frequency.

#### 3.5.2 | Working with noninteger frequencies

As stated in Section 2.1, one of the assumptions (3) was that all stimulation frequencies are integers. Here, we aim to relax this assumption.

From Section 3, we know that this integer frequency assumption determines whether the “coefficient search for linear combinations” step can be solved with LDE solvers. Therefore, this problem becomes as follows:

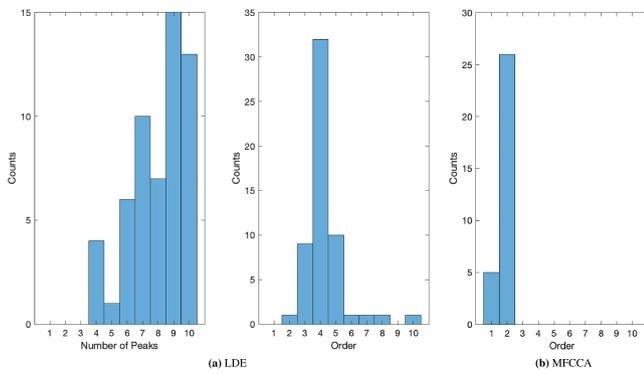
Solve for  $\alpha'x + \beta'y = \gamma'$ , where  $x$  and  $y$  are unknown integers,  $\alpha', \beta', \gamma' \in \mathbb{R}$ .

One way to convert the problem back to integer space is to multiply both sides of the equation by  $10^z$ , where  $z$  is the largest decimal place in  $\alpha', \beta'$ , and  $\gamma'$ . After applying this multiplication to both sides, this could then be solved by the algorithms described above. Steps involved in this process are as follows:

1. In “frequency peak identification,” find peaks to the frequency resolution of  $10^{-z}$  Hz with  $\epsilon \times 10^{-z}$  Hz tolerance, where  $z$  is the largest decimal place in the frequency candidates.
2. In “coefficient search for linear combinations,” multiply both the identified peaks and the candidate frequencies by  $10^z$  to convert everything to integers.
3. Follow the steps explained in Sections 3.2 and 3.3.
4. By the end of “output generation,” convert the output back to decimals by multiplying  $10^{-z}$ .

## 4 | VALIDATION DATASET

In this work, we used previously collected data [16] to evaluate the effectiveness of the proposed LDE decoder. The dataset contains SSVEPs recorded by EEG from nine healthy participants (four females and five males, aged 22–33 years, mean 26.8 years, standard deviation 3.7 years). EEG was collected using g.USBamp amplifier with g.SAHARA dry electrodes (g.tec medical engineering GmbH, Austria) from six channels over the visual cortex



**FIGURE 3** Histogram of most common settings for highest accuracies decoded with 1- to 30-s data. [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

(PO3, POz, PO4, O1, Oz, O2, international 10-10 system) at 512 Hz, and 50 Hz notch filter and 0.5 to 60 Hz band-pass filter applied in the g.USBamp settings during data acquisition.

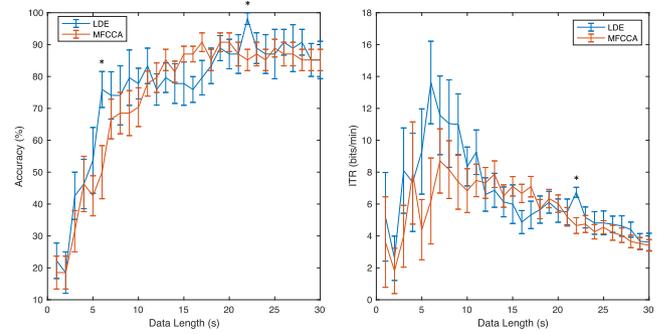
Data from each participant include 30 s multifrequency SSVEPs stimulated by six frequency pairs ( $6 \times 30$  s) using frequency superposition stimulation method with OR logic on 50% duty cycle square waves [16]. The six frequency pairs are as follows: 7 & 9 Hz, 7 & 11 Hz, 7 & 13 Hz, 9 & 11 Hz, 9 & 13 Hz, and 11 & 13 Hz. Stimulation was delivered by an Adafruit NeoPixel  $8 \times 8$  RGB LED panel (70 mm  $\times$  70 mm) in red color, located 100 cm in front of the participant, centered to the eyes.

**TABLE 1** Accuracy of linear Diophantine equation decoder at different settings with 30 s data. Highest accuracy is highlighted in green.

Accuracy \ Order Number of peaks	1	2	3	4	5	6	7	8	9	10
1	16.67%	25.93%	31.48%	31.48%	31.48%	31.48%	31.48%	31.48%	31.48%	31.48%
2	31.48%	48.15%	66.67%	70.37%	70.37%	70.37%	70.37%	70.37%	70.37%	70.37%
3	38.89%	53.70%	77.78%	83.33%	83.33%	85.19%	87.04%	87.04%	87.04%	87.04%
4	38.89%	55.56%	85.19%	88.89%	87.04%	81.48%	83.33%	85.19%	85.19%	85.19%
5	40.74%	59.26%	85.19%	85.19%	83.33%	75.93%	72.22%	75.93%	75.93%	75.93%
6	42.59%	64.81%	94.44%	94.44%	81.48%	77.78%	74.07%	75.93%	75.93%	75.93%
7	48.15%	64.81%	88.89%	88.89%	81.48%	79.63%	79.63%	79.63%	81.48%	83.33%
8	46.30%	59.26%	85.19%	85.19%	81.48%	81.48%	79.63%	77.78%	77.78%	77.78%
9	48.15%	59.26%	90.74%	85.19%	83.33%	83.33%	83.33%	81.48%	81.48%	85.19%
10	50.00%	59.26%	90.74%	88.89%	81.48%	85.19%	85.19%	83.33%	81.48%	87.04%

**TABLE 2** Accuracy of multifrequency canonical correlation analysis decoder at different settings with 30 s data. Highest accuracy is highlighted in green.

Order	1	2	3	4	5	6	7	8	9	10
Accuracy	70.37%	85.19%	81.48%	77.78%	70.37%	59.26%	44.44%	16.67%	16.67%	16.67%

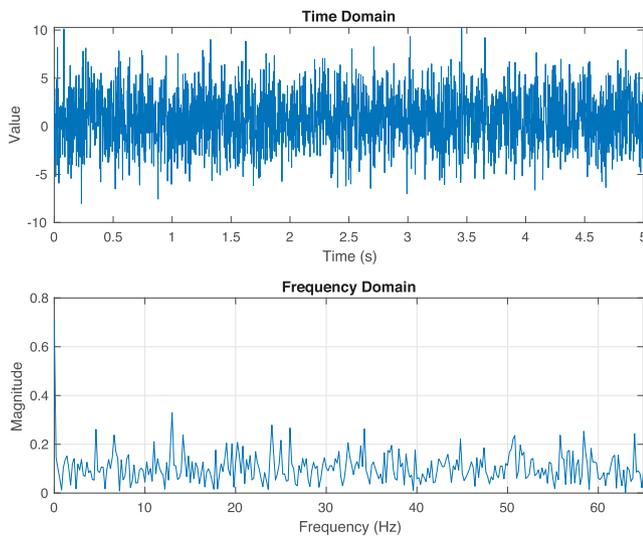


**FIGURE 4** Accuracy and information transfer rate comparison between linear Diophantine equation (LDE) and multifrequency canonical correlation analysis (MFCCA). Error bars show standard errors of the average accuracies from each subject. \* labels significant difference at 5% with Wilcoxon signed-rank test. [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

This study was approved by the University of Melbourne Human Research Ethics Committee (Ethics ID 1851283). Written consent was received from all subjects prior to the experiments.

## 5 | LDE PERFORMANCE

This validation dataset meets the assumptions on multifrequency SSVEP in Section 2.1 [18], and the integer candidate frequency set is also known from the experimental protocol. Table 1 summarizes the average LDE decoding accuracies across all participants at different LDE settings



**FIGURE 5** Example of test signal used in testing run times of linear Diophantine equation (LDE) and multifrequency canonical correlation analysis (MFCCA). [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

with full 30-s EEG recordings. Number of peaks  $n$  determines how many peaks  $p$  are extracted from the “frequency peak identification” step that makes the vector of peaks  $\mathbf{P}$  and order bounds the search range  $[\underline{c}, \bar{c}]$  in “output generation,” where, in this particular case,  $\underline{c} = 1$  and  $\bar{c} = \text{order}$ . As a comparison, Table 2 lists the average decoding accuracy from MFCCA with different order settings with the same 30-s EEG recordings. Cells highlighted in green in both tables indicate the respective highest accuracies in the two tables. From the tables, we can see that the highest accuracy from LDE is higher than that from MFCCA. This demonstrates the potential of LDE as a multifrequency SSVEP decoding algorithm.

In order to understand the best operating range of the decoder setting for the LDE decoder (number of peaks and order), LDE decoding was done with different data length starting from 1 to 30 s with 1 s increments and with the same range of decoder settings as in Table 1 (1 to 10 for both number of peaks and order). The highest accuracy was then identified at each data length and the corresponding decoder settings were recorded and summarized into Figure 3a. Note that in cases where multiple settings led to the same highest accuracy, such as in Table 1, both setting were included. The same was done for MFCCA as a comparison, as shown in Figure 3b, where it can be seen that, in this dual-frequency case, the common LDE settings that may result in the best decoding outcome are identifying nine peaks from the recorded EEG data and searching for coefficients up to order 4 ( $\bar{c} = 4$ ). On the other hand, MFCCA performs the best in general at order 2. This implies that MFCCA mainly uses frequency inter-

**TABLE 3** Run time (s) comparison between LDE and MFCCA when iteratively running the algorithms multiple times.

Run time (s)		
Number of runs	LDE	MFCCA
10	0.0146	0.0816
100	0.1231	0.6234
1000	1.2404	5.8813
10000	12.3759	59.0758
Settings		
LDE	Order	4
	Number of peaks	9
MFCCA	Order	2
	Data length	5s
	Number of channels in test signal	1

Abbreviations: LDE, linear Diophantine equation; MFCCA, multifrequency canonical correlation analysis.

**TABLE 4** Run time (s) comparison between LDE and MFCCA with different data length (s) used in decoding.

Run time (s)		
Data length (s)	LDE	MFCCA
5	0.1244	0.5740
10	0.1247	1.1363
20	0.1287	2.1113
30	0.1363	3.0192
Settings		
LDE	Order	4
	Number of peaks	9
MFCCA	Order	2
	Number of channels in test signal	1
	Number of runs	100

Abbreviations: LDE, linear Diophantine equation; MFCCA, multifrequency canonical correlation analysis.

action information up to order 2, whereas LDE further utilizes information at higher orders.

Figure 4 shows comparisons between LDE and MFCCA accuracy and information transfer rate (ITR) with different data length at the optimal common setting identified above from Figure 3a. The error bars represent standard error of the average accuracies from each subject. The Wilcoxon signed-rank test was performed to test between LDE and MFCCA accuracies at each data length and significant differences at 5% were labeled with \* in the figure. The ITR is calculated as

$$ITR = \frac{60}{T} \left[ \log_2 N + p_{acc} \log_2 p_{acc} + (1 - p_{acc}) \log_2 \left( \frac{1 - p_{acc}}{N - 1} \right) \right], \quad (8)$$

where  $T$  is the time window (seconds) for a trial or time needed to produce one result,  $N$  is the number of targets or possible choices, and  $p_{acc}$  is the mean accuracy [26].

Simulated run time tests were also done with an artificially generated test signal that mimicked the recorded

**TABLE 5** Run time (s) comparison between LDE and MFCCA with different numbers of channels in the test signal used in decoding.

Run time (s)		
Number of channels	LDE	MFCCA
1	0.1519	0.6583
8	0.1525	0.8238
16	0.1336	1.0339
32	0.1322	3.3695
64	0.1491	10.1371
128	0.1841	25.0981
Settings		
LDE	Order	4
	Number of peaks	9
MFCCA	Order	2
	Data length	5s
Number of runs		100

Abbreviations: LDE, linear Diophantine equation; MFCCA, multifrequency canonical correlation analysis.

**TABLE 6** Run time (s) comparison between LDE and MFCCA when decoding at different orders.

Run time (s)		
Order	LDE	MFCCA
1	0.1272	0.1671
2	0.1199	0.5931
3	0.1263	1.2793
4	0.1239	2.4192
5	0.1213	3.9164
6	0.1240	11.2753
7	0.1204	24.4093
8	0.1359	37.5039
9	0.1223	57.2156
10	0.1205	83.8450
Settings		
LDE	Number of peaks	9
	Data length	5s
	Number of channels in test signal	1
	Number of runs	100

Abbreviations: LDE, linear Diophantine equation; MFCCA, multifrequency canonical correlation analysis.

multifrequency SSVEP. This signal was generated by performing frequency superposition [16] of 11 and 13 Hz with OR then adding white Gaussian noise at  $-10$  dB signal-to-noise ratio (SNR), similar to the wide-band SNR in recorded SSVEP [27]. The signal was sampled at 512 Hz, which is consistent with the sampling rate of EEG recording used in our previous experiments. Figure 5 shows an example of the test signal in both time domain and frequency domain.

The run times of the two algorithms were then tested with the test signal and results are shown in Table 3.

**TABLE 7** Run time (s) of LDE when decoding with different number of peaks.

Run time (s)		
Number of peaks	LDE	
10	0.1379	
20	0.2436	
30	0.3591	
50	0.5970	
100	1.1724	
Settings		
LDE	Order	4
	Data length	5s
	Number of channels in test signal	1
	Number of runs	100

Abbreviations: LDE, linear Diophantine equation.

Results show an “almost linear” trend with small overhead processes also contributing to the measured run times.

Run time comparison between LDE and MFCCA were also done with varying data length used in decoding (Table 4) and number of channels in the data (Table 5). How run times of the two algorithms change as order in decoder settings increases are shown in Table 6. For LDE, the run time as number of peaks in decoder setting increases is also shown in Table 7. All tests were run on a MacBook Pro 13-inch 2017 with 2.5 GHz dual-core Intel Core i7 processor and 16 GB 2133 MHz RAM. All unnecessary processes and applications are closed leaving only MATLAB (The MathWorks, Inc., USA) running the decoding of test signal.

Table 3 shows that the run time of LDE is almost one fifth of the run time of MFCCA in the identified optimal setting (LDE: order 4, number of peaks 9; MFCCA: order 2) with 5-s simplified single-channel data. It is recognized that the system response time is an important factor in human–computer interactions and can affect human performance [28], keeping the response time low would be critical for real-time/online applications. In BCI, the decoding algorithm run time determines the lower bound of system response time. The longest acceptable response time in human–computer interaction varies with the complexity of the task, but for typical BCI use cases where the response is usually activation or simple feedback, the response time should be kept lower than 200 ms [29, 30]. This means that in an online BCI application, LDE could be easily applied to complex scenarios with over 100 targets, while MFCCA would fail with only around 30 targets.

In Tables 4–6, we can see that the changes in LDE run times are almost negligible as data length, number of channels, and order increases. In comparison, MFCCA shows a clearer and faster increase in run time as these parameters increases. Table 7 shows that LDE run time is more sensitive to number of peaks compared to the other parameters.

From the results above, we can see that LDE performed at least equally well as MFCCA with a general common decoder setting. The computational cost of LDE ( $O(n \log n)$ ) was much lower than MFCCA ( $O(n^3)$ ), as explained in Section 1. This made LDE a suitable candidate for online decoding of multifrequency SSVEP especially in complex settings, for example, when the order of interest is high or there are more frequencies coded in a single target.

## 6 | CONCLUSION

In this paper, a novel multifrequency steady-state visual evoked potential (SSVEP) decoding algorithm based on LDE solvers was introduced, which has a theoretically lower time complexity ( $O(n \log n)$ ) compared to multifrequency canonical correlation analysis (MFCCA) ( $O(n^3)$ ). Experimental results showed that the LDE performance was as good as MFCCA in terms of decoding accuracy and information transfer rate even though the run time of LDE is only one fifth of that of MFCCA. This makes LDE a preferred algorithm in real-time applications where instant response and timely feedback is critical.

Future work could investigate potential methods to systematically determine the best selection of parameters in the LDE decoder and to further improve the multifrequency SSVEP decoding accuracy.

## AUTHOR CONTRIBUTIONS

**Jing Mu:** Conceptualization, data curation, formal analysis, investigation, methodology, project administration, software, visualization, writing—original draft.

**Ying Tan:** Conceptualization, formal analysis, methodology, supervision, writing—review and editing.

**David B. Grayden:** Conceptualization, formal analysis, methodology, supervision, writing—review and editing.

**Denny Oetomo:** Conceptualization, formal analysis, funding acquisition, methodology, supervision, writing—review and editing.

## CONFLICT OF INTEREST STATEMENT

There is no conflict of interest.

## ORCID

Jing Mu  <https://orcid.org/0000-0002-3289-2002>

Ying Tan  <https://orcid.org/0000-0001-8495-0246>

David B. Grayden  <https://orcid.org/0000-0002-5497-7234>

Denny Oetomo  <https://orcid.org/0000-0002-2680-6489>

## REFERENCES

1. D. Regan, *Human brain electrophysiology: Evoked potentials and evoked magnetic fields in science and medicine*, Elsevier, 1989.
2. T. O. Zander, C. Kothe, S. Welke, and M. Rötting, *Utilizing secondary input from passive brain-computer interfaces for enhancing human-machine interaction*, International Conference on Foundations of Augmented Cognition. Springer, 2009, pp. 759–771.
3. L. F. Nicolas-Alonso and J. Gomez-Gil, *Brain computer interfaces, a review*, *Sensors* **12** (2012), no. 2, 1211–1279.
4. H.-J. Hwang, J.-H. Lim, Y.-J. Jung, H. Choi, S. W. Lee, and C.-H. Im, *Development of an SSVEP-based BCI spelling system adopting a QWERTY-style LED keyboard*, *J. Neurosci. Methods* **208** (2012), no. 1, 59–65.
5. G. R. Müller-Putz and G. Pfurtscheller, *Control of an electrical prosthesis with an SSVEP-based BCI*, *IEEE Trans. Biomed. Eng.* **55** (2008), no. 1, 361–364.
6. R. Ortner, B. Z. Allison, G. Korisek, H. Gaggl, and G. Pfurtscheller, *An SSVEP BCI to control a hand orthosis for persons with tetraplegia*, *IEEE Trans. Neural Syst. Rehabilitation Engineering* **19** (2011), no. 1, 1–5.
7. C. J. Perera, I. Naoitunna, C. Sadaruwan, R. A. R. C. Gopura, and T. D. Lalitharatne, *SSVEP based BMI for a meal assistance robot*, 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, 2016, pp. 002295–002300.
8. X. Chen, Y. Wang, M. Nakanishi, X. Gao, T.-P. Jung, and S. Gao, *High-speed spelling with a noninvasive brain-computer interface*, *Proc. Natl. Acad. Sci.* **112** (2015), no. 44, E6058–E6067.
9. X. Chen, B. Zhao, Y. Wang, S. Xu, and X. Gao, *Control of a 7-DOF robotic arm system with an SSVEP-based BCI*, *Int. J. Neural Syst.* **28** (2018), no. 08, 1850018.
10. S. M. T. orres Müller, W. C. Celeste, T. F. Bastos-Filho, and M. Sarcinelli-Filho, *Brain-computer interface based on visual evoked potentials to command autonomous robotic wheelchair*, *J. Med. Biol. Eng.* **30** (2010), no. 6, 407–415.
11. C. S. Herrmann, *Human EEG responses to 1–100 Hz flicker: Resonance phenomena in visual cortex and their potential correlation to cognitive phenomena*, *Exper. Brain Res.* **137** (2001), no. 3–4, 346–353.
12. M. Xu, F. He, T.-P. Jung, X. Gu, and D. Ming, *Current challenges for the practical application of electroencephalography-based brain-computer interfaces*, *Engineering* **7** (2021), no. 12, 1710–1712.
13. M. H. Chang, H. J. Baek, S. M. Lee, and K. S. Park, *An amplitude-modulated visual stimulation for reducing eye fatigue in SSVEP-based brain-computer interfaces*, *Clinical Neurophysiol.* **125** (2014), no. 7, 1380–1391.
14. X. Chen, Z. Chen, S. Gao, and X. Gao, *Brain-computer interface based on intermodulation frequency*, *J. Neural Eng.* **10** (2013), no. 6, 066009.
15. H.-J. Hwang, D. H. Kim, C.-H. Han, and C.-H. Im, *A new dual-frequency stimulation method to increase the number of visual stimuli for multi-class SSVEP-based brain-computer interface (BCI)*, *Brain Res.* **1515** (2013), 66–77.
16. J. Mu, D. B. Grayden, Y. Tan, and D. Oetomo, *Frequency superposition—A multi-frequency stimulation method in SSVEP-based BCIs*, 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). IEEE, 2021, pp. 5924–5927.
17. K.-K. Shyu, P.-L. Lee, Y.-J. Liu, and J.-J. Sie, *Dual-frequency steady-state visual evoked potential for brain computer interface*, *Neurosci. Lett.* **483** (2010), no. 1, 28–31.
18. J. Mu, Y. Tan, D. B. Grayden, and D. Oetomo, *Multi-frequency canonical correlation analysis (MFCCA): A generalised decoding algorithm for multi-frequency SSVEP*, 2021 43rd Annual Inter-

- national Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). IEEE, 2021, pp. 6151–6154.
19. L. Liang, J. Lin, C. Yang, Y. Wang, X. Chen, S. Gao, and X. Gao, *Optimizing a dual-frequency and phase modulation method for SSVEP-based BCIs*, *J. Neural Eng.* **17** (2020), no. 4, 046026.
  20. Z. Lin, C. Zhang, W. Wu, and X. Gao, *Frequency recognition based on canonical correlation analysis for SSVEP-based BCIs*, *IEEE Trans. Biomed. Eng.* **53** (2006), no. 12, 2610–2614.
  21. N. Rasiwasia, D. Mahajan, V. Mahadevan, and G. Aggarwal, *Cluster canonical correlation analysis*, *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, 2014, pp. 823–831.
  22. O. Friman, I. Volosyayk, and A. Graser, *Multiple channel detection of steady-state visual evoked potentials for brain-computer interfaces*, *IEEE Trans. Biomed. Eng.* **54** (2007), no. 4, 742–750.
  23. Z. Tabanfar, F. Ghassemi, and M. H. Moradi, *Estimating brain periodic sources activities in steady-state visual evoked potential using local fourier independent component analysis*, *Biomed. Sig. Proc. Control* **71** (2022), 103162.
  24. W. J. Gilbert and A. Pathria, *Linear Diophantine equations*, 1990. preprint.
  25. A. Schrijver, *Theory of linear and integer programming*, John Wiley & Sons, 1998.
  26. J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, *Brain-computer interfaces for communication and control*, *Clinical Neurophysiol.* **113** (2002), no. 6, 767–791.
  27. B. Liu, X. Huang, Y. Wang, X. Chen, and X. Gao, *Beta: A large benchmark database toward SSVEP-BCI application*, *Front. Neurosci.* **14** (2020), 627.
  28. I. S. MacKenzie and C. Ware, *Lag as a determinant of human performance in interactive systems*, *Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing Systems*, 1993, pp. 488–493.
  29. C. Attig, N. Rauh, T. Franke, and J. F. Krems, *System latency guidelines then and now—Is zero latency really considered necessary?* *International Conference on Engineering Psychology and Cognitive Ergonomics*. Springer, 2017, pp. 1–14.
  30. R. B. Miller, *Response time in man-computer conversational transactions*, *Proceedings of the December 9–11, 1968, Fall Joint Computer Conference, Part I*, 1968, pp. 267–277.

## AUTHOR BIOGRAPHIES



**Jing Mu** received her BEng degree in Automotive Engineering from the University of Shanghai for Science and Technology, Shanghai, China, in 2014 and her MEng degree (with distinction) in Mechatronics from the University of Melbourne, Victoria, Australia, in 2017, where she is pursuing her PhD degree in brain-computer interface with the Department of Mechanical Engineering. She is currently a Research Fellow with the Department of Biomedical Engineering, University of Melbourne, and part of the ARC Training Centre in Cognitive Computing for

Medical Technologies. Her current research interests include noninvasive brain-computer interfaces (BCIs), assistive technologies, and robotics.



**Ying Tan** is a Professor in the Department of Mechanical Engineering at The University of Melbourne, Australia. She received her bachelor's degree from Tianjin University, China, in 1995, and her PhD from the National University of Singapore in 2002. She joined McMaster University in 2002 as a postdoctoral fellow in the Department of Chemical Engineering. Since 2004, she has been with the University of Melbourne. She was awarded an Australian Postdoctoral Fellow (2006–2008) and a Future Fellow (2009–2013) by the Australian Research Council. She is Fellow of the Institute of Electrical and Electronic Engineers (FIEEE), Fellow of the Institution of Engineers of Australia (FIEAUST), and Fellow of Asia-Pacific Artificial Intelligence Association. Her research interests are in intelligent systems, nonlinear systems, real-time optimization, sampled-data systems, rehabilitation robotic systems, human motor learning, wearable sensors, and model-guided machine learning.



**David B. Grayden** received his PhD degree in Electrical and Electronic Engineering from The University of Melbourne, Australia in 1999. He is now Professor and Clifford Chair in Neural Engineering at the Department of Biomedical Engineering, The University of Melbourne. His main research interests are in understanding how the brain processes information, how best to present information to the brain using medical bionics, such as the bionic ear and bionic eye, and how to record information from the brain, such as for brain-computer interfaces and for epileptic seizure prediction.



**Denny Oetomo** received his BEng from the Australian National University and PhD in Robotics from the National University of Singapore (2004). He is currently Professor in the Department of Mechanical Engineering at the University of Melbourne. His research interests are in the topics of robot dynamics and its interaction strategies with the task and human dynamics.

**How to cite this article:** J. Mu, Y. Tan, DB. Grayden, and D. Oetomo, *Linear Diophantine equation (LDE) decoder: A training-free decoding algorithm for multifrequency SSVEP with reduced computation cost*, *Asian J. Control* (2023), 1–13. <https://doi.org/10.1002/asjc.3050>