

RADBOUD UNIVERSITY NIJMEGEN



---

# Steps towards a continuous visual evoked code modulated Brain Computer Interface

---

BACHELOR THESIS ARTIFICIAL INTELLIGENCE

*Author:*

Maren Zahn  
s4825489

July 2020

*Supervisors*

Prof. Dr. Peter Desain  
Dr. Ceci Verbaarschot  
Dr. Sara Ahmadi

## Abstract

This thesis investigated ways to make visual evoked code modulated brain computer interfaces (BCIs) continuous, as this would extend the usage of BCIs. In an offline experiment two approaches of making the stimuli continuous were tested. The first method modified the sequences digitally; every location got a binary sequence which was a mix of two Gold codes at the left and right. The other approach mixed the colors of the two Gold codes at the edges of the screen. The trials were classified based on the activation of either all all location or only the activation of the extreme left and right. The digitally modified sequences classified with the activation for all locations reached the same accuracy as the control. However, it had a smaller average distance between the target location and the predicted location. This method is thus an improvement of continuous BCIs.

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Methods</b>	<b>5</b>
2.1	Participants . . . . .	5
2.2	Equipment . . . . .	5
2.3	Procedure of the experiment . . . . .	5
2.4	Stimuli . . . . .	5
2.4.1	Digital stochastic . . . . .	6
2.4.2	Color gradient . . . . .	7
2.4.3	Control . . . . .	7
2.4.4	Targets . . . . .	8
2.5	Analysis . . . . .	8
2.5.1	Preprocessing . . . . .	8
2.5.2	Classifier . . . . .	8
2.5.3	Tuning curve . . . . .	8
2.5.4	2 class mapping . . . . .	9
2.6	Measuring performance . . . . .	9
<b>3</b>	<b>Results</b>	<b>9</b>
3.0.1	Standard classification . . . . .	9
3.0.2	Tuning curve . . . . .	9
3.0.3	2 class mapping . . . . .	10
<b>4</b>	<b>Discussion</b>	<b>10</b>
4.1	Classifier with grey values . . . . .	10
4.2	Digital stochastic better than color gradient . . . . .	11
4.3	2 class mapping . . . . .	11
4.4	Confidences . . . . .	12
4.5	The ITR . . . . .	13
4.6	Improvements and further research . . . . .	13
<b>5</b>	<b>Conclusion</b>	<b>13</b>
<b>6</b>	<b>Acknowledgements</b>	<b>14</b>
<b>7</b>	<b>Appendix</b>	<b>16</b>
7.1	Sequences for digital stochastic . . . . .	16
7.2	Sequences for color gradient . . . . .	18
7.3	Sequences for control . . . . .	21

# 1 Introduction

A Brain Computer Interface (BCI) is a system measuring brain activity and decoding this activity to the user's intentions. The output can be used as an alternative form of communication. This is convenient for users who have limited or no control over their muscles. The BCI serves as a translator between someone's brain activity and intentions. For locked-in patients this means that they can still communicate because no muscle movement is needed, though BCIs are also used to improve or even supplement motor control.

In visual BCIs the brain activity is evoked by visual stimulation. While the user is looking at the stimulus, the stimulus leads to Visual Evoked Potentials (VEPs) in the visual cortex. Electrodes placed on the outside of the visual cortex pick up these VEPs via Electroencephalography (EEG).

In BCIs based on code modulated VEP (c-VEPs) the stimuli flicker between black and white according to some sequences. The presentation has different stimuli on the screen, each stimulus is a possible target and has its own sequence. Gold codes can be used as sequences. Gold codes are a set of pseudo random binary sequences with small cross correlation to the other sequences. These sequences consist rising (01) and falling edges (10), both having different responses in the visual cortex. In the training phase, the BCI learns the response to these edges. The recorded brain activity can be decomposed into these two responses. This decomposition is then used to reconstruct the sequence which evoked the brain activity. The decoded sequence can be matched to the target. An advantage of this approach is that even the response to an unseen sequence can be predicted. Hence, the user does not need to train on all the different sequences.

A classic example of a BCI used for communication, is the speller. In this setup, every letter has its own button on the screen which all flicker with a different sequence. The letter which sequence is most likely to have evoked the brain activity is the output. c-VEP BCIs have the best performance for communication, compared to other types of BCIs [1]. Other types of VEP-BCIs focus on the temporal or frequency features of the VEPs. The drawback of c-VEP BCIs is that they require training, whereas the other types do not necessarily require training [1]. Performance is assessed with the Information Transfer Rate (ITR), a conventional measurement to compare BCIs. The ITR includes the accuracy, the speed and the number of possibilities to choose from.

If the target at which the user is looking is in the central focus area, it is called overt attention. This means that eye movement is involved to get the selected target in the central focus area. If the user has its attention in his/ her periphery, it is called covert attention. This form of selecting does not require eye-movement and might thus be more suitable for completely locked-in patients. Research shows that BCIs can be operated via covert attention, but it is still more difficult than overt [8, 9]. A study comparing a BCI with an eye tracker found that a user preferred a BCI because it required less precise eye movements and can also be used covertly [4]. This paper uses overt attention, later it could be tested with covert attention.

Little research has been done to a more continuous form. The above discussed BCIs all have a discrete number of targets. And every target has its own button on the screen with the corresponding code. If these buttons are too close to each other, the performance decreases [6]. This limits the functionality of the current BCIs; with a certain number of buttons, only a certain number of actions can be performed. This paper investigates two quasi continuous alternatives. For simplicity,

this paper is focused on one dimension: from left to right.

One dimensional continuous output can be used to control any one dimensional value. Some example applications are: volume control, speed control in a game or adapting brightness. This is very practical for users depending on BCIs, but also for healthy users because they can control something by just looking. Figure 1 is an example of volume control: a sliding bar between low and high volume can be moved to the desired location if the output of the BCI gives this location. The red box would contain the stimulus which evokes the brain activity.

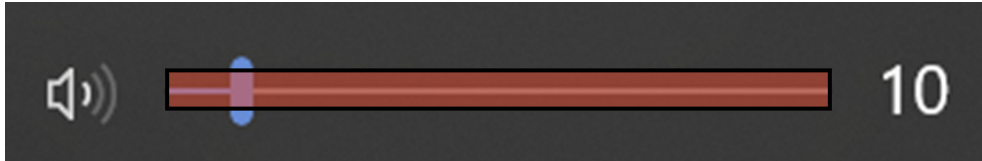


Figure 1: Using continuous output to control volume. The red box over the slider presents the stimulus.

Extending the output to two dimensions would massively increase the functionality of visual BCIs. They would provide the same functionality as a cursor, which also uses the selected location. The continuous flicker could be implemented into any computer program, enabling the users to have the same control as a cursor. Which would mean that locked-in patients are no longer limited to BCI specific software.

A completely continuous stimulus is impossible if the presentation is on a screen with a discrete number of pixels. Dividing the screen into sections with the size of a pixel, would result in a big number of sections. Because this is the first attempt to making a continuous stimulus, a smaller number of sections is used. When the user looks at the screen, a group of pixels falls within his/her central attention area. In this paper the screen is divided into sections that have the size of the functional field of view of the user. All the sections are placed directly next to each other, horizontally aligned. Like in discrete BCIs, every section still has its own sequence. But for the continuous presentation, these sequences are correlated. The closer the sections are to each other, the higher the correlation. If the response to one section gets mixed with the response to the neighbor, it is not really a problem anymore because the response is probably similar as well.

The assignment of sequences to each section such that neighbors are similar is done via two methods. The first is a digital stochastic morph. This means that the Gold codes are modified to become a mix. Two Gold codes are placed at the edges of the screen, the sections in between are a mix of these codes at the edges. This is done stochastically per bit; the chance of a section getting the same bit as the Gold code on the left, depends on how far that section is. This results in a mix between the two. Figure 2 is a visualisation of how the mix could look.

The second method also places two Gold codes at the edges of the screen. Here, the mix is made from the colors. So the middle sections form a gradient going from the color on the left, morphing into the color on the right. This means that there are 4 possible stimuli, see figure 3. When the colors at the edges do not have the same color, the sections in between will have different shades of grey. [3] has shown that BCIs can work with 5 different shades of grey.

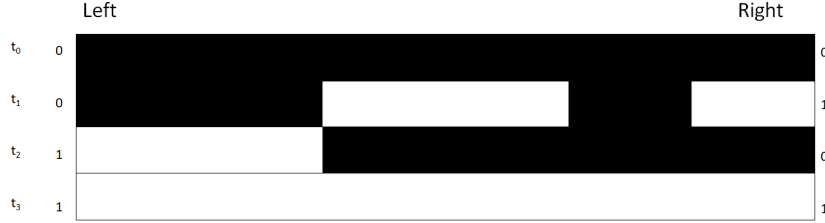


Figure 2: An example of the stimulus at 4 time points according to the digital stochastic morph.

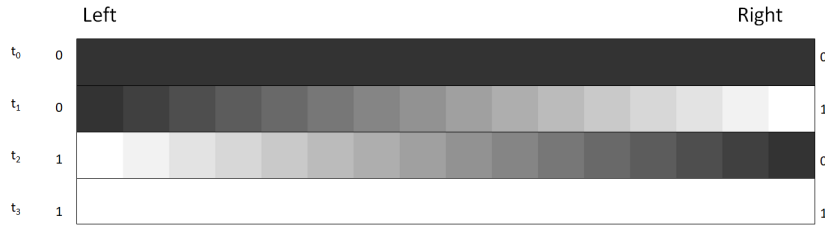


Figure 3: An example of the stimulus at 4 time points according to the digital stochastic morph.

The decoding needs to be able to extract the attended location from the recorded brain activity. This is done in two different ways.

The first approach is based on tuning curves. Neighboring sections have similar sequences and thus also similar responses. So the neighbors of the target section will also have high activation, just like with tuning curves. The classifier takes the activation of the neighbors into account when making a decision.

The other approach uses the fact that every section is a mix of the edges. The recorded response is compared to the response to the sequences at the edges. The ratio of similarity between left and right is then mapped to the predicted location.

Is it possible to design a more continuous visual code modulated BCI with stimuli close to each other? Such a BCI would need to perform better than a BCI which has stimuli directly next to each other with all different sequences. Performance is measured in accuracy, ITR and the average distance of the predicted section and actual section. An offline analysis is used to determine the performance of the BCI. Offline analysis means that the data is recorded first and analyzed later. This gives flexibility in trying different decoding methods. The performance of the digital stochastic morph and the color gradient will be compared to a control group. The control group has all the sections directly next to each other and every section has its own Gold code. The performance of the two new conditions is computed via the tuning curve method and via the edges ratio approach.

The performance of the digital stochastic morph and the color gradient with the tuning curve decoding are expected to be higher compared to the control group. The classifier can base its decision on more information because the neighbors have influence as well.

The performance of the two conditions with the edges-ratio approach is also expected to be

higher than the control’s performance, since this decoding method is designed for correlated sequences.

## 2 Methods

### 2.1 Participants

Due to measures regarding the COVID-19 crisis using participants was somewhat limited. Still, 3 participants participated in the experiment, including the author. Their average age is 22 ( $\pm 2.1$ ), all participants are female and have normal or corrected to normal vision. They gave consent in accordance with the Donders Centre for Cognition consent form. Participants were able to quit the experiment at any time without reason. The collected data was stored anonymously.

### 2.2 Equipment

The monitor used to present the stimuli to the participants has a refresh rate of 75 Hz. The monitor has 1920 pixels (width) by 1080 pixels (height).

To measure the brain activity to store this and the stimulus data, a BCI kit from MindAffect [5] is used. This kit contains six electrodes, including one ground and one reference electrode. It measures with a frequency of 200 Hz.

### 2.3 Procedure of the experiment

At the start of the experiment the participants received information about the purpose and protocol of the experiment. The participants sit comfortably 100 cm from the monitor. A trial consists of a cueing period and a stimulus period. In the cueing period, the target is indicated by highlighting the target area green. This takes 1 second. After the cue there is a variable time interval between 300 and 700 milliseconds to ensure that the response used for classification is not the response to the cue. Then the stimulus period starts, in which all sections flicker. The stimulus period takes about 4 seconds, in these 4 seconds the sequence is repeated twice. After the stimulation period all targets turn grey for 2 seconds and then the next trial starts. 15 consecutive trials form a block. In the upper right corner the number of completed blocks is displayed. After a block the participant can take a break of maximal 2 minutes to rest. After 2 minutes, or if the participant is ready sooner, the next block can start. Every participant completes 15 blocks. A block has a condition type (digital stochastic, color gradient or control) and the order of the conditions is randomly determined. The task for the participants during the experiment is to keep looking at the indicated position, preferably without blinking during the stimulation period.

### 2.4 Stimuli

All 3 conditions have some overlapping attributes, the aim is to keep that as high as possible. In this section the common features are discussed, in the other subsection the specifics for each condition are examined. A distance of 100 cm between the monitor and the participant means that the function visual field of a participant on the monitor is approximately 3.5 cm horizontally [7]. For the used monitor, this means that the presentation can have 16 sections. Every section is 3.4 cm, which corresponds to 120 pixels.

While making example presentations, it turned out that fixating on a screen that is flickering all over is quite difficult. To make this easier, the height of the flicker is limited to 4 cm, 108 pixels. Every section has a black non-flickering fixation cross in the middle to ensure that the participant looks at the middle. If the continuous stimulus would be used in a real application and not just an experiment, the screen will also contain non-flickering elements, like letters. The fixation crosses mimic this as well.

The frequency of the presentation is 60 Hz, thus the next bit will be displayed after 17 ms approximately. Every sequence has a length of 126 bits because the MindAffect kit comes with a set of Gold codes with this length. With the frequency of 60 Hz this means that displaying one sequence takes about 2.1 seconds.

The background of the presentation is black, so to make the stimuli stand out, the darkest color is dark-grey instead of completely black. Figure 4 is a screen capture of the presentation during the color gradient condition where all features discussed above can be seen.

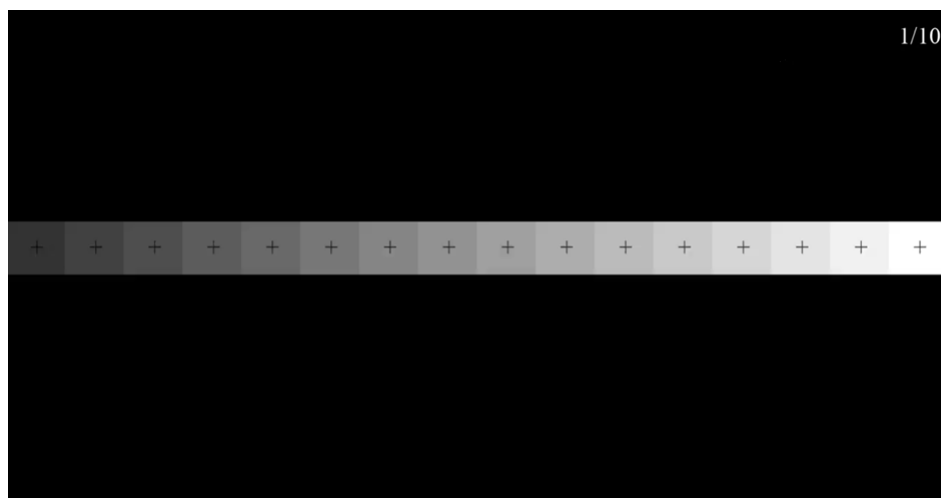


Figure 4: An example of the presentation of the color gradient condition: 16 blocks, each a different shade of grey, with a fixation cross in the middle.

#### 2.4.1 Digital stochastic

Two Gold codes are selected for the extreme left and right section. Every section of the 14 sections in between have a probability of receiving the a bit from the left or the right Gold code. This probability distribution is linear in the location. For the probability for each section, see table 1.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.93	0.87	0.8	0.73	0.67	0.6	0.53	0.47	0.4	0.34	0.27	0.2	0.13	0.07	0
0	0.07	0.13	0.20	0.27	0.34	0.40	0.47	0.53	0.60	0.67	0.73	0.80	0.87	0.93	1

Table 1: Probability distribution for each section. Row 1: probabilities of selecting a bit from the Gold code from the left Row 2: probabilities of selecting a bit from the Gold code from the right

Figure 5 is a visualization of a slice of the mix. Blue means that the bit comes from the left, red from the right.

One important feature of a sequence is that it should not contain the same bit more than twice after each other. If a code would have more than twice a zero or one after each other, the interval between flashes or the duration of a flash would be too long. A way to avoid this in any sequence, is to temporarily replace the zeros in the binary sequence by -1. Multiply each bit by  $[-1,1]$ , convert the -1 back to 0 and the result is a binary sequence where a bit does not repeat more than twice after each other. The new sequence has double the length of the original sequence.

After forming the sequences for the 14 sections in the middle, the sequences do not have the property of not repeating the same value more than twice. To achieve that, the previously described modulation is applied. To still end up with a sequence of 126 bits, the starting Gold codes need to have a length of 63. These Gold codes and the sequences for the other sections can be found in the appendix, section 7.1. Note that in the visualization in figure 5 the modulation has not been applied yet.

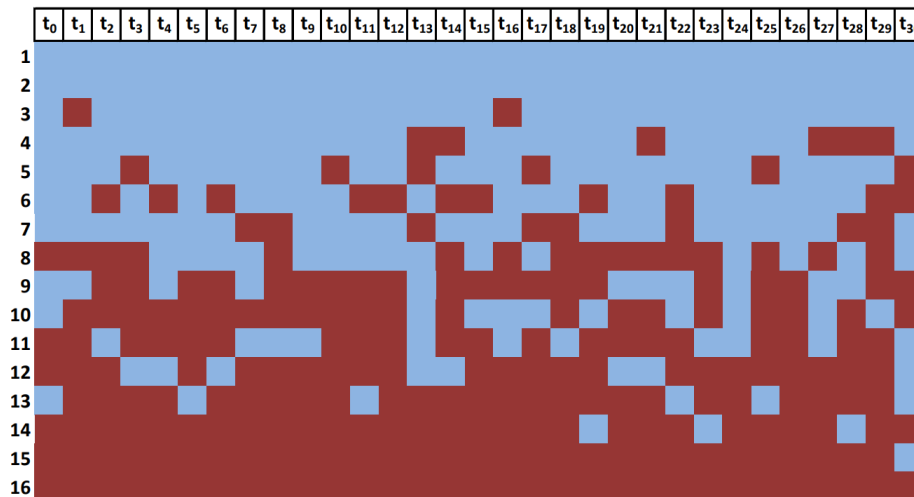


Figure 5: A visualization of a slice of the mix used before modulation. Blue means that the bit comes from the left, red indicates a bit from the right. Horizontally is the time, vertically the different sections.

### 2.4.2 Color gradient

The Gold codes selected and the middle sequences computed for this condition can be found in the appendix, section 7.2.

### 2.4.3 Control

The sequences used for this condition come from the set of Gold codes that were provided by the MindAffect kit. The sequences can be found in the appendix, section 7.3.



#### 2.4.4 Targets

To provide enough trials for each condition and each section, a subsection of the sections is selected as possible targets. The two sections at the far left and right are in the selected subset because they are needed to for the 2-class mapping decoding. The section in the middle is selected because it is an even blend of the two sections at the sides. Section 15 is selected to include two sections directly next to each other. And section 4 serves as a an example of a section that is not at the edge and not in the middle. In table 2 shows the selected sections in red and also the order of the sections.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Table 2: The red numbers show the selection of sections as possible targets.

## 2.5 Analysis

### 2.5.1 Preprocessing

Before analysis the frequencies in the EEG-data below 1 Hz and above 25 Hz are removed by a filter. The EEG-signal is then downsampled to 67 Hz, the stimuli information gets upsampled from 60 Hz to 67 Hz.

### 2.5.2 Classifier

The MindAffect kit comes with a standard classifier. It can be used on the continuous stimuli; every section becomes a class. The result of the classification process is a probability for every class that it is the target. The class with the highest probability is the predicted target. The classifier uses multiple canonical correlation analysis (CCA) as a spatial filter. The CCA learns for every type of stimulus event the average response. It then finds a spatial filter to maximize this representation. It has been shown that using CCA for c-VEP BCIs can improve the performance [2]. By default, the kit works with rising and falling edges. It thus learns the response to a change from 0 to 1, and from 1 to 0. This is used for the control and the digital stochastic condition. Rising and falling edges can not be used for the color gradient condition because that contains other values than 0 and 1. For this condition the event types are the grey-values. Hence here the CCA learns the average response to all of the 16 grey-scales.

### 2.5.3 Tuning curve

The certainties for every class are taken into account when determining the attended section. A support vector machine (SVM) with a one-degree polynomial kernel function uses all 16 probabilities to compute the predicted class. The SVM already has a function to get the confidence for each prediction.

The experiment only tested a subset of the sections, so there is no response to some of the grey values, but all 16 probabilities are needed. To make up for this, the missing probabilities are interpolated from the know probabilities. This is only the case for the color gradient condition. All probabilities for the digital stochastic are known because the only event types are rising and falling edges.

### 2.5.4 2 class mapping

This method only takes the probabilities of the standard classifier for the extreme classes as input. These two numbers are then mapped to the predicted location. The mapping is performed by a SVM with a one-degree polynomial kernel function which gets trained on the probabilities for left and right and emits the predicted class.

The trials are expected to form a diagonal line; if the probability for left is high, the probability for right should be low and vice versa. The distance of a trial from this diagonal line forms the confidence of the classifier that it is the correct decision. The diagonal is the regression line from the data points.

## 2.6 Measuring performance

The first way to analyze the performance of any decoding methods is the accuracy: how often does the model predict the correct section? The information transfer rate can be calculated once the accuracy is known. There are 16 sections so there are 16 possible options. Every trial has a stimulation time of 4 seconds and an average inter-trial time of 2.7 seconds thus the formula for the ITR in bits per minute is:

$$ITR = \log_2(N) + P \log_2(1 - P) \log_2\left(\frac{1 - P}{N - 1}\right) * 60/6.7 \quad (1)$$

In the continuous cases some errors are worse than others. The absolute difference between the actual section and the predicted section form the difference distance. For now, it is expressed as the number of sections from the correct section, it could be multiplied by the width of a section to express it in centimeters. This measurement reflects the degree of incorrectness.

Every prediction comes with a confidence of that prediction being correct. A good model should have a higher confidence for the correct decision. The last measurement if is there is a significant difference between the confidences for correct or incorrect decisions.

## 3 Results

All of the results below are the averages after 10 fold cross validation.

### 3.0.1 Standard classification

The first step is to compare the two experimental conditions with the control while using the standard classifier. These results are displayed in table 3. A star means that there is a significant difference according to a one-sided Wilcoxon signed rank test ( $p < 0.05$ ) compared to the control. Both the digital stochastic and the color gradient condition perform worse in terms of accuracy and ITR compared to the control. The difference distance is not significantly lower. The confidence for the correct predictions is for all conditions not higher than the confidences for the incorrect predictions.

### 3.0.2 Tuning curve

The results for the two conditions with the tuning curve are in table 4. Compared to the same condition but with the standard classifier, all measurements improved significantly, except for the

	<b>Control</b>	<b>Digital stochastic</b>	<b>Color gradient</b>
<b>Accuracy</b>	0.64 ( $\pm 0.2$ )	0.39 ( $\pm 0.12$ )*	0.33 ( $\pm 0.06$ )*
<b>ITR</b>	15.98 ( $\pm 8.94$ )	6.21 ( $\pm 3.68$ )*	6.50 ( $\pm 4.81$ )*
<b>Difference distance</b>	2.33 ( $\pm 1.32$ )	1.70 ( $\pm 0.45$ )	2.55 ( $\pm 0.86$ )
<b>Confidence correct higher</b>	No	No	No

Table 3: Measurements for standard classification. A star means a significant difference with control.

difference distance of the color gradient. Significant differences compared with the standard classifier are marked with a triangle.

These results can also be compared to the control. The color gradient compared to the control shows that the color gradient does significantly worse for accuracy and ITR.

The digital stochastic condition’s accuracy and ITR are not higher or lower than the control. However, the difference distance is significantly lower in the digital stochastic control compared to the control. Significant difference (better or worse) compared with the control are again indicated with a star.

Again, the confidences do not show a difference for the correct or incorrect predictions.

	<b>Digital Stochastic</b>	<b>Color Gradient</b>
<b>Accuracy</b>	0.65 ( $\pm 0.08$ ) $^{\Delta}$	0.50 ( $\pm 0.09$ ) $^{\Delta}$ *
<b>ITR</b>	15.29 ( $\pm 3.96$ ) $^{\Delta}$	9.58 ( $\pm 3.13$ ) $^{\Delta}$ *
<b>Difference distance</b>	1.02 ( $\pm 0.27$ ) $^{\Delta}$ *	2.85 ( $\pm 0.88$ )
<b>Confidence correct higher</b>	No	No

Table 4: Measurements for tuning curve.  $\Delta$  means a significant difference compared to the standard classifier. A \* indicates a significant difference compared to the control.

### 3.0.3 2 class mapping

The results of the 2 class mapping are shown in table 5. These can be compared to the results of the standard classifier. Both conditions shows a significant improvement in accuracy, and thus also ITR. The difference distances are not significantly better.

There is no significant difference between the conditions.

Compared to the control, both conditions do not perform better. The control even does significantly better according to accuracy and ITR. The digital stochastic condition does have a significantly lower difference distance. No difference in the confidences are measured.

## 4 Discussion

### 4.1 Classifier with grey values

Using grey values instead of just black and white gives different responses than the responses that are usually obtained. It is interesting to compare them. Figure 6 shows the response to rising (0

	Digital stochastic	Color gradient
Accuracy	0.52 ( $\pm 0.08$ ) $^{\Delta}$ *	0.46 ( $\pm 0.11$ ) $^{\Delta}$ *
ITR	10.26 ( $\pm 2.95$ ) $^{\Delta}$ *	8.18 ( $\pm 4.17$ ) $^{\Delta}$ *
Difference distance	1.63 ( $\pm 0.32$ )*	2.62 ( $\pm 1.03$ )
Average distance to diagonal	No	No

Table 5: Measurements for 2 class classification mapping.  $^{\Delta}$  means a significant difference compared to standard classifier. \* means a significant difference compared to control.

to 1) and falling (1 to 0) edges. Figure 7 shows the learned responses to some of the grey values. Here it becomes visible that an even mix of black and white (e.g. 0.47 or 0.53), does not give the same response as the sum of the responses to black and white (0 and 1).

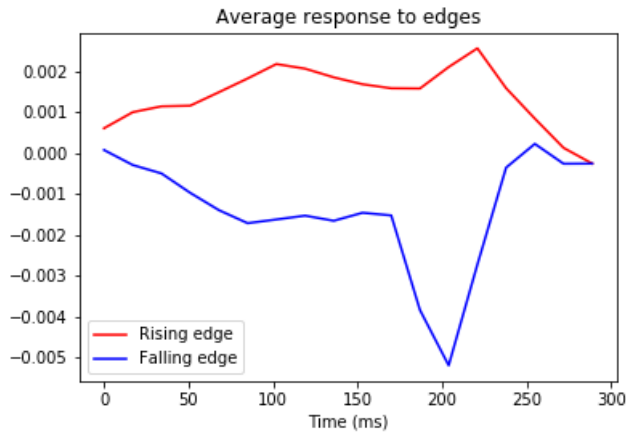


Figure 6: The learned responses to rising and falling edges.

## 4.2 Digital stochastic better than color gradient

The fact that the response to grey is not a mix of the response to black and white might be the explanation of why the digital stochastic gives better results than the color gradient (with tuning curve decoding). In the digital stochastic condition, the sections in the middle evoke the same response types as the section at the left and right. Only the order and frequency of these two responses is different. In the color gradient, the responses types are different. The mixed sections are not a mix of the responses to the two extremes.

## 4.3 2 class mapping

No previous research has tried to an approach like the 2 class mapping, so a visual check to see if the sections can be distinguished in a plane of the confidence of left and right can be helpful. In figure 8 this is plotted. There seems to be some division between the classes. This is in accordance

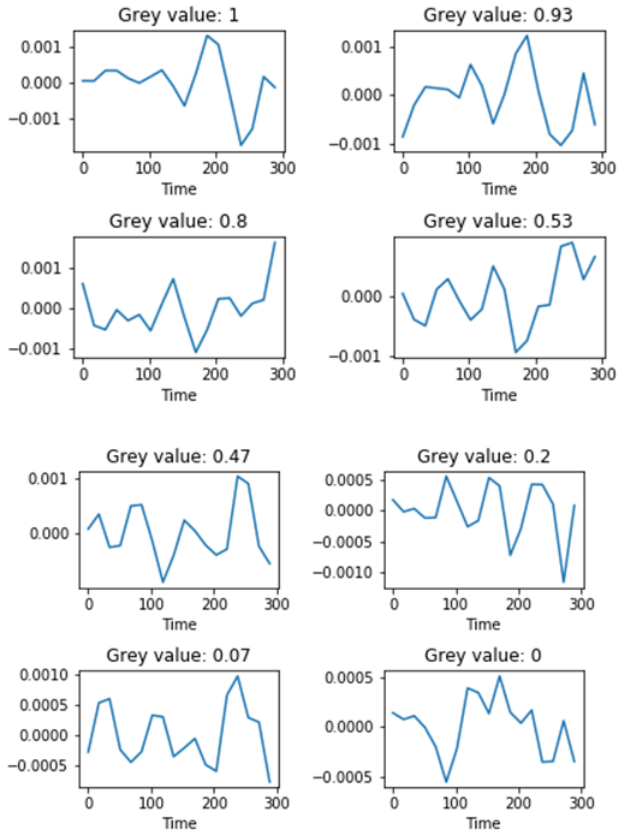


Figure 7: The learned responses to some of the grey values.

with finding that the 2 class mapping method improved performance compared to the standard classifier.

#### 4.4 Confidences

The confidence measure according to the distance to the diagonal is a new method. To illustrate how it could be used, see figure 9. Here the confidences for the data points in the green zone would be considered high enough. This is just an example and the actual application of using confidences would depend on the context in which it is used.

None of the classification methods had a reliable confidence measure, even the control. A different method to determine the confidence would be helpful in the future.

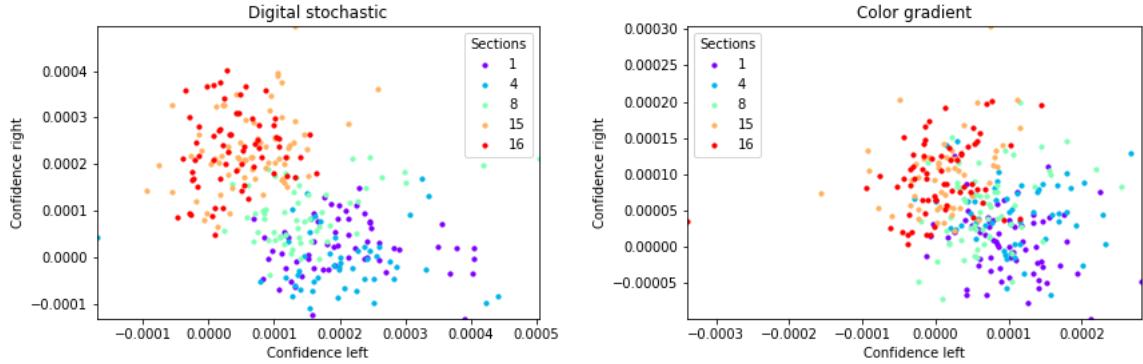


Figure 8: The division of sections in a plane with the confidences for left and right. Every color is a different section.

## 4.5 The ITR

The ITR is computed based on 16 sections. However, all the classifiers only predicted 5 classes. So the ITR should not directly be compared to the ITR of other BCIs. The classifier learned that it should only predict 5 classes so it did not make a true decision between 16 classes. This does not mean that the classifiers gave the other classes zero probability, just never the highest.

## 4.6 Improvements and further research

A first thing that could be improved is making the sections smaller. The smaller the sections, the more continuous the BCI is and that is after all the goal. Even if multiple sections would evoke a mix of potentials, the tuning curve analysis would probably be able to handle this. With more sections the accuracy could go down, but the ITR could increase. If the error distance would be low, a lower accuracy is not necessarily bad.

Testing every section in the experiment would make the classification probably better. The confidences for every section will be more accurate which could lead to better overall accuracy.

Another improvement would be to try it with covert attention, since the main target population of BCIs is people with decreased muscle control. Even the people who can not move their eyes would be able to use it.

Alternatives for the decoding could be investigated in future research. A SVM is chosen for this research because it does not need parameters. But fitting a function through the confidences for each class could maybe also work if the correct function is chosen.

Last but not least, is the user experience. The pleasantness of the stimuli is not considered here but this is an important aspect.

## 5 Conclusion

Only the digital stochastic condition in combination with the tuning curve decoding method performs better than the control: it has a lower difference distance and the same accuracy and ITR. This method is thus an improvement for continuous BCIs because it makes the same number

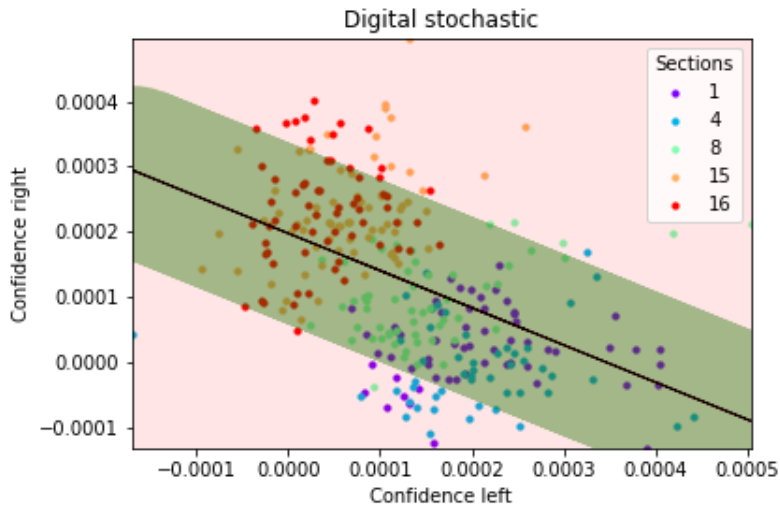


Figure 9: An example of defining confidences based on distance to diagonal. All data points in the green range would get a high confidence, red low.

of correct predictions but if it makes an incorrect prediction, it is closer to the target than the control. It is thus possible to design a visual BCI with stimuli close to each other and maintaining performance. However, the improvement is only for the difference distance and not for the accuracy and ITR.

Tuning curves and the 2 class mapping have shown to be a way to improve the performance of classifying correlated responses compared to the standard classification of correlated responses.

## 6 Acknowledgements

Thanks to Jason Farquhar from MindAffect for helping me with the code from the kit!

## References

- [1] G. Bin et al. “VEP-based brain-computer interfaces: time, frequency, and code modulations [Research Frontier]”. In: *IEEE Computational Intelligence Magazine* 4.4 (2009), pp. 22–26.
- [2] Guangyu Bin et al. “A high-speed BCI based on code modulation VEP”. In: *Journal of neural engineering* 8 (Mar. 2011), p. 025015. DOI: 10.1088/1741-2560/8/2/025015.
- [3] Felix Gemblar et al. “Five Shades of Grey: Exploring Quintary m -Sequences for More User-Friendly c-VEP-Based BCIs”. In: *Computational Intelligence and Neuroscience* 2020 (Mar. 2020), pp. 1–11. DOI: 10.1155/2020/7985010.
- [4] Ivo Käthner, Andrea Kübler, and Sebastian Halder. “Comparison of eye tracking, electrooculography and an auditory brain-computer interface for binary communication: A case study with a participant in the locked-in state”. In: *Journal of neuroengineering and rehabilitation* 12 (Sept. 2015), p. 76. DOI: 10.1186/s12984-015-0071-z.
- [5] *MindAffect*. <http://mindaffect.nl>. 2020.
- [6] K. B. Ng, A. P. Bradley, and R. Cunnington. “Effect of competing stimuli on SSVEP-based BCI”. In: *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 2011, pp. 6307–6310.
- [7] Keith Rayner. *Eye movements in reading: Perceptual and language processes*. Elsevier, 2012.
- [8] Matthias Treder, Nico Schmidt, and B Blankertz. “Gaze-independent brain-computer interfaces based on covert attention and feature attention”. In: *Journal of neural engineering* 8 (Dec. 2011), p. 066003. DOI: 10.1088/1741-2560/8/6/066003.
- [9] Dan Zhang et al. “An independent brain-computer interface using covert non-spatial visual selective attention”. In: *Journal of neural engineering* 7.1 (2010), p. 016010.



## 7 Appendix

### 7.1 Sequences for digital stochastic

```
[0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 1]
[1 1 1 1 1 1 1 0 1 1 0 0 1 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 1 0 0 1 1 1 1 0 1 1 1 1]
[1 1 1 1 0 1 1 0 0 0 0 1 0 0 0 0]
[1 1 1 1 1 0 1 1 1 0 0 1 0 0 0 0]
[0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 1]
[0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1]
[1 1 1 1 1 1 1 1 0 0 0 0 1 0 0 0]
[1 1 1 1 1 0 1 1 0 0 0 1 0 0 0 0]
[0 0 0 0 0 1 0 0 1 1 1 0 1 1 1 1]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 1]
[1 1 1 1 1 1 1 1 1 0 0 1 0 0 0 0]
[1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0]
[0 0 0 0 1 0 0 0 1 1 1 1 1 1 1 1]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[1 1 1 1 1 0 1 1 0 1 0 0 0 0 0 0]
[0 0 0 0 0 1 0 0 1 0 1 1 1 1 1 1]
[0 0 1 0 0 0 0 1 1 0 0 1 1 1 1 1]
[1 1 0 1 1 1 1 0 0 1 1 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[1 1 1 0 1 1 1 0 1 0 0 1 0 0 0 0]
```

[0 0 0 1 0 0 0 1 0 1 1 0 1 1 1 1]  
[0 0 0 0 0 1 1 1 0 0 1 1 0 1 1 1]  
[1 1 1 1 1 0 0 0 1 1 0 0 1 0 0 0]  
[0 0 0 0 0 0 0 1 1 1 0 1 1 0 1 1]  
[1 1 1 1 1 1 1 0 0 0 1 0 0 1 0 0]  
[1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0]  
[0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1]  
[0 0 0 0 1 0 0 1 1 1 1 1 1 0 1 1 1]  
[1 1 1 1 0 1 1 0 0 0 0 0 1 0 0 0]  
[1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0]  
[0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1]  
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]  
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]  
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]  
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]  
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]  
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]  
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]  
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]  
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  
[1 1 1 1 1 0 1 1 0 1 1 0 1 0 1 0 0 0]  
[0 0 0 0 0 1 0 0 1 0 0 1 0 1 1 1 1 1]  
[0 0 0 0 0 0 1 1 1 0 1 1 1 1 1 1 1 1]  
[1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0]  
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]  
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]  
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  
[1 1 1 1 1 0 1 0 1 0 1 0 0 1 0 0 0 0]  
[0 0 0 0 0 1 0 1 0 1 1 0 1 1 1 1 1 1]  
[0 0 0 0 0 0 0 1 0 1 0 1 1 1 1 1 1 1]  
[1 1 1 1 1 1 1 1 0 1 0 1 0 0 0 0 0 0]  
[1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 0]  
[0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 0]  
[0 0 0 0 1 1 0 0 1 0 0 1 1 1 1 1 1 1]  
[1 1 1 1 0 0 1 1 0 1 1 0 0 0 0 0 0 0]  
[1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0]  
[0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1]  
[0 0 0 0 1 0 0 0 1 1 0 1 1 1 1 1 1 1]  
[1 1 1 1 0 1 1 1 1 0 0 1 0 0 0 0 0 0]  
[0 0 1 1 0 1 0 0 0 1 1 0 1 1 1 1 1 1]  
[1 1 0 0 1 0 1 1 1 0 0 1 0 0 0 0 0 0]  
[1 1 1 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0]

```

[0 0 0 1 0 1 1 1 1 1 0 0 1 1 1 1 1]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[0 1 1 0 0 0 0 0 1 1 0 1 1 1 1 1]
[1 0 0 1 1 1 1 1 0 0 1 0 0 0 0 0]
[1 1 1 1 1 0 0 0 1 0 0 0 0 1 0 0]
[0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[0 0 0 0 0 0 1 1 1 1 1 0 1 1 1 1]
[1 1 1 1 1 1 0 0 0 0 0 1 0 0 0 0]
[1 1 1 1 1 0 1 0 0 0 1 1 0 0 0 0]
[0 0 0 0 0 1 0 1 1 1 0 0 1 1 1 1]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[1 1 1 1 1 1 1 1 0 0 0 0 1 0 0 0]
[0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1]
[0 0 0 0 0 0 1 0 0 1 1 1 1 0 1 1]
[1 1 1 1 1 1 0 1 1 1 0 0 0 0 1 0 0]
references

```

## 7.2 Sequences for color gradient

```

1.00 0.93 0.87 0.80 0.73 0.67 0.60 0.53 0.47 0.40 0.33 0.27 0.20 0.13 0.07 0.00
0.00 0.07 0.13 0.20 0.27 0.33 0.40 0.47 0.53 0.60 0.67 0.73 0.80 0.87 0.93 1.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

```







[0. 1. 1. 0. 1. 0. 1. 1. 1. 1. 0. 0. 0. 1. 0. 0.]  
[1. 0. 0. 1. 0. 1. 0. 0. 0. 0. 1. 1. 1. 0. 1. 1.]  
[1. 1. 0. 0. 1. 0. 1. 0. 0. 0. 0. 1. 1. 1. 1. 1.]  
[0. 0. 1. 1. 0. 1. 0. 1. 1. 1. 1. 0. 0. 0. 0. 0.]  
[1. 0. 0. 1. 1. 0. 1. 1. 1. 1. 1. 0. 0. 0. 1. 0.]  
[0. 1. 1. 0. 0. 1. 0. 0. 0. 0. 0. 1. 1. 1. 0. 1.]  
[1. 1. 0. 0. 1. 1. 0. 0. 1. 1. 1. 0. 0. 0. 0. 1.]  
[0. 0. 1. 1. 0. 0. 1. 1. 0. 0. 0. 1. 1. 1. 1. 0.]  
[0. 1. 1. 0. 0. 1. 1. 1. 1. 1. 1. 1. 0. 0. 0. 0.]  
[1. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0. 1. 1. 1. 1.]  
[1. 0. 1. 1. 0. 0. 1. 0. 0. 1. 1. 1. 1. 0. 0. 0.]  
[0. 1. 0. 0. 1. 1. 0. 1. 1. 0. 0. 0. 0. 1. 1. 1.]  
[0. 0. 1. 0. 0. 1. 1. 0. 0. 1. 0. 0. 0. 0. 1. 1.]  
[1. 1. 0. 1. 1. 0. 0. 1. 1. 0. 1. 1. 1. 1. 0. 0.]  
[1. 1. 1. 0. 1. 1. 0. 0. 0. 1. 0. 1. 1. 1. 0. 0.]  
[0. 0. 0. 1. 0. 0. 1. 1. 1. 0. 1. 0. 0. 0. 1. 1.]  
[1. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 1.]  
[0. 1. 1. 1. 0. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 0.]  
[0. 1. 0. 0. 0. 1. 0. 0. 1. 0. 1. 0. 0. 0. 0. 0.]  
[1. 0. 1. 1. 1. 0. 1. 1. 0. 1. 0. 1. 1. 1. 1. 1.]  
[1. 1. 0. 1. 1. 1. 0. 0. 1. 0. 1. 0. 1. 1. 1. 1.]  
[0. 0. 1. 0. 0. 0. 1. 1. 0. 1. 0. 1. 0. 0. 0. 0.]  
[0. 1. 1. 0. 1. 1. 1. 0. 1. 1. 0. 1. 0. 1. 1. 1.]  
[1. 0. 0. 1. 0. 0. 0. 1. 0. 0. 1. 0. 1. 0. 0. 0.]  
[0. 1. 0. 0. 1. 0. 0. 0. 1. 0. 0. 1. 0. 1. 0. 0.]  
[1. 0. 1. 1. 0. 1. 1. 1. 0. 1. 1. 0. 1. 0. 1. 1.]  
[0. 1. 0. 1. 1. 0. 1. 1. 0. 0. 1. 1. 0. 1. 1. 1.]  
[1. 0. 1. 0. 0. 1. 0. 0. 1. 1. 0. 0. 1. 0. 0. 0.]  
[1. 1. 0. 1. 0. 0. 1. 1. 0. 1. 1. 1. 0. 1. 1. 0.]  
[0. 0. 1. 0. 1. 1. 0. 0. 1. 0. 0. 0. 1. 0. 0. 1.]  
[1. 0. 0. 1. 0. 1. 1. 1. 1. 1. 0. 1. 0. 1. 1. 0.]  
[0. 1. 1. 0. 1. 0. 0. 0. 0. 0. 1. 0. 1. 0. 0. 1.]  
[1. 0. 1. 1. 0. 1. 0. 0. 1. 0. 0. 0. 0. 1. 1. 0.]  
[0. 1. 0. 0. 1. 0. 1. 1. 0. 1. 1. 1. 1. 0. 0. 1.]  
[0. 0. 1. 0. 0. 1. 0. 1. 1. 0. 1. 0. 1. 1. 1. 0.]  
[1. 1. 0. 1. 1. 0. 1. 0. 0. 1. 0. 1. 0. 0. 0. 1.]  
[1. 0. 0. 1. 0. 0. 1. 0. 1. 1. 0. 0. 0. 1. 0. 1.]  
[0. 1. 1. 0. 1. 1. 0. 1. 0. 0. 1. 1. 1. 0. 1. 0.]  
[0. 0. 1. 1. 0. 1. 1. 0. 0. 0. 0. 0. 1. 1. 0. 1.]  
[1. 1. 0. 0. 1. 0. 0. 1. 1. 1. 1. 1. 0. 0. 1. 0.]  
[1. 1. 1. 0. 0. 1. 0. 1. 0. 1. 1. 1. 1. 0. 1. 1.]  
[0. 0. 0. 1. 1. 0. 1. 0. 1. 0. 0. 0. 0. 1. 0. 0.]  
[0. 1. 1. 1. 0. 0. 1. 0. 1. 0. 1. 0. 1. 1. 0. 1.]  
[1. 0. 0. 0. 1. 1. 0. 1. 0. 1. 0. 1. 0. 0. 1. 0.]  
[1. 1. 0. 0. 0. 1. 1. 0. 0. 0. 1. 0. 1. 0. 1. 1.]  
[0. 0. 1. 1. 1. 0. 0. 1. 1. 1. 0. 1. 0. 1. 0. 0.]

[1. 1. 1. 0. 0. 0. 1. 1. 1. 0. 0. 0. 0. 1. 0. 1.]  
 [0. 0. 0. 1. 1. 1. 0. 0. 0. 1. 1. 1. 1. 0. 1. 0.]  
 [0. 1. 1. 1. 0. 0. 0. 1. 0. 1. 0. 0. 0. 0. 0. 0.]  
 [1. 0. 0. 0. 1. 1. 1. 0. 1. 0. 1. 1. 1. 1. 1. 1.]  
 [1. 0. 1. 1. 1. 0. 0. 0. 1. 0. 1. 1. 0. 0. 1. 0.]  
 [0. 1. 0. 0. 0. 1. 1. 1. 0. 1. 0. 0. 1. 1. 0. 1.]  
 [1. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 1. 1. 0.]  
 [0. 1. 0. 1. 1. 1. 0. 1. 1. 1. 0. 0. 1. 0. 0. 1.]  
 [1. 1. 0. 1. 0. 0. 0. 0. 1. 0. 0. 1. 1. 0. 1. 1.]  
 [0. 0. 1. 0. 1. 1. 1. 1. 0. 1. 1. 0. 0. 1. 0. 0.]  
 [1. 1. 1. 0. 1. 0. 0. 1. 0. 1. 0. 0. 1. 1. 1. 1.]  
 [0. 0. 0. 1. 0. 1. 1. 0. 1. 0. 1. 1. 0. 0. 0. 0.]  
 [1. 1. 1. 1. 0. 1. 0. 1. 0. 0. 1. 1. 0. 1. 0. 1.]  
 [0. 0. 0. 0. 1. 0. 1. 0. 1. 1. 0. 0. 1. 0. 1. 0.]  
 [0. 1. 1. 1. 1. 0. 1. 1. 1. 0. 0. 0. 1. 0. 1. 0.]  
 [1. 0. 0. 0. 0. 1. 0. 0. 0. 1. 1. 1. 0. 1. 0. 1.]  
 [1. 1. 0. 0. 0. 0. 1. 1. 0. 0. 1. 1. 1. 0. 0. 0.]  
 [0. 0. 1. 1. 1. 1. 0. 0. 1. 1. 0. 0. 0. 1. 1. 1.]  
 [0. 1. 1. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 1. 1. 0.]  
 [1. 0. 0. 1. 1. 1. 1. 0. 1. 1. 1. 1. 0. 0. 0. 1.]  
 [0. 1. 0. 0. 1. 1. 1. 0. 0. 1. 1. 0. 1. 0. 1. 0.]  
 [1. 0. 1. 1. 0. 0. 0. 1. 1. 0. 0. 1. 0. 1. 0. 1.]  
 [0. 1. 0. 1. 1. 0. 0. 0. 1. 1. 0. 1. 1. 0. 1. 0.]  
 [1. 0. 1. 0. 0. 1. 1. 1. 0. 0. 1. 0. 0. 1. 0. 1.]  
 [1. 0. 1. 0. 1. 1. 0. 1. 1. 1. 1. 0. 1. 1. 1. 1.]  
 [0. 1. 0. 1. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0. 0. 0.]  
 [1. 1. 0. 1. 0. 1. 1. 0. 0. 1. 1. 0. 0. 1. 0. 1.]  
 [0. 0. 1. 0. 1. 0. 0. 1. 1. 0. 0. 1. 1. 0. 1. 0.]  
 [0. 0. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 1. 1. 0. 1.]  
 [1. 1. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 0. 0. 1. 0.]  
 [1. 0. 0. 0. 1. 0. 1. 1. 1. 0. 1. 0. 1. 1. 0. 0.]  
 [0. 1. 1. 1. 0. 1. 0. 0. 0. 1. 0. 1. 0. 0. 1. 1.]  
 [0. 0. 1. 1. 1. 0. 1. 0. 0. 0. 1. 1. 1. 0. 0. 1.]  
 [1. 1. 0. 0. 1. 1. 1. 1. 0. 0. 0. 0. 1. 1. 0. 0.]  
 [0. 0. 0. 1. 1. 0. 0. 1. 0. 1. 1. 0. 1. 0. 0. 1.]  
 [1. 1. 1. 0. 0. 1. 1. 0. 1. 0. 0. 1. 0. 1. 1. 0.]  
 [0. 0. 0. 0. 1. 1. 0. 0. 0. 0. 1. 1. 0. 1. 0. 0.]  
 [1. 1. 1. 1. 0. 0. 1. 1. 1. 1. 0. 0. 1. 0. 1. 1.]  
 [0. 1. 1. 1. 1. 0. 0. 0. 0. 1. 1. 0. 0. 1. 1. 1.]  
 [1. 0. 0. 0. 0. 1. 1. 1. 1. 0. 0. 1. 1. 0. 0. 0.]  
 [1. 0. 1. 1. 1. 1. 0. 1. 1. 0. 1. 0. 0. 0. 0. 1.]  
 [0. 1. 0. 0. 0. 0. 1. 0. 0. 1. 0. 1. 1. 1. 1. 0.]



[1. 1. 0. 1. 1. 1. 1. 1. 0. 1. 0. 0. 0. 0. 1. 0.]  
[0. 0. 1. 0. 0. 0. 0. 0. 1. 0. 1. 1. 1. 1. 0. 1.]  
[0. 0. 0. 1. 0. 0. 0. 0. 0. 1. 0. 0. 1. 1. 1. 0.]  
[1. 1. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 0. 0. 0. 1.]  
[1. 1. 1. 1. 0. 1. 1. 0. 1. 1. 0. 1. 1. 0. 0. 0.]  
[0. 0. 0. 0. 1. 0. 0. 1. 0. 0. 1. 0. 0. 1. 1. 1.]  
[0. 0. 0. 0. 0. 1. 0. 1. 0. 0. 0. 1. 0. 0. 1. 1.]  
[1. 1. 1. 1. 1. 0. 1. 0. 1. 1. 1. 0. 1. 1. 0. 0.]  
[0. 1. 1. 1. 1. 1. 0. 1. 0. 1. 1. 1. 0. 1. 0. 0.]  
[1. 0. 0. 0. 0. 0. 1. 0. 1. 0. 0. 0. 1. 0. 1. 1.]  
[0. 0. 1. 1. 1. 1. 1. 1. 0. 0. 1. 0. 1. 0. 1. 0.]  
[1. 1. 0. 0. 0. 0. 0. 0. 1. 1. 0. 1. 0. 1. 0. 1.]  
[0. 0. 0. 1. 1. 1. 1. 1. 1. 0. 0. 1. 0. 1. 1. 1.]  
[1. 1. 1. 0. 0. 0. 0. 0. 0. 1. 1. 0. 1. 0. 0. 0.]  
[0. 0. 0. 0. 1. 1. 1. 1. 1. 1. 0. 1. 1. 0. 0. 1.]  
[1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 1. 1. 0.]  
[0. 0. 0. 0. 0. 1. 1. 0. 1. 1. 1. 1. 1. 1. 1. 0.]  
[1. 1. 1. 1. 1. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1.]  
[0. 0. 0. 0. 0. 0. 1. 1. 1. 1. 1. 0. 1. 1. 0. 1.]  
[1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 1. 0. 0. 1. 0.]