BIOMEDICAL ENGINEERING GROUP
UNIVERSITY OF VALLADOLID

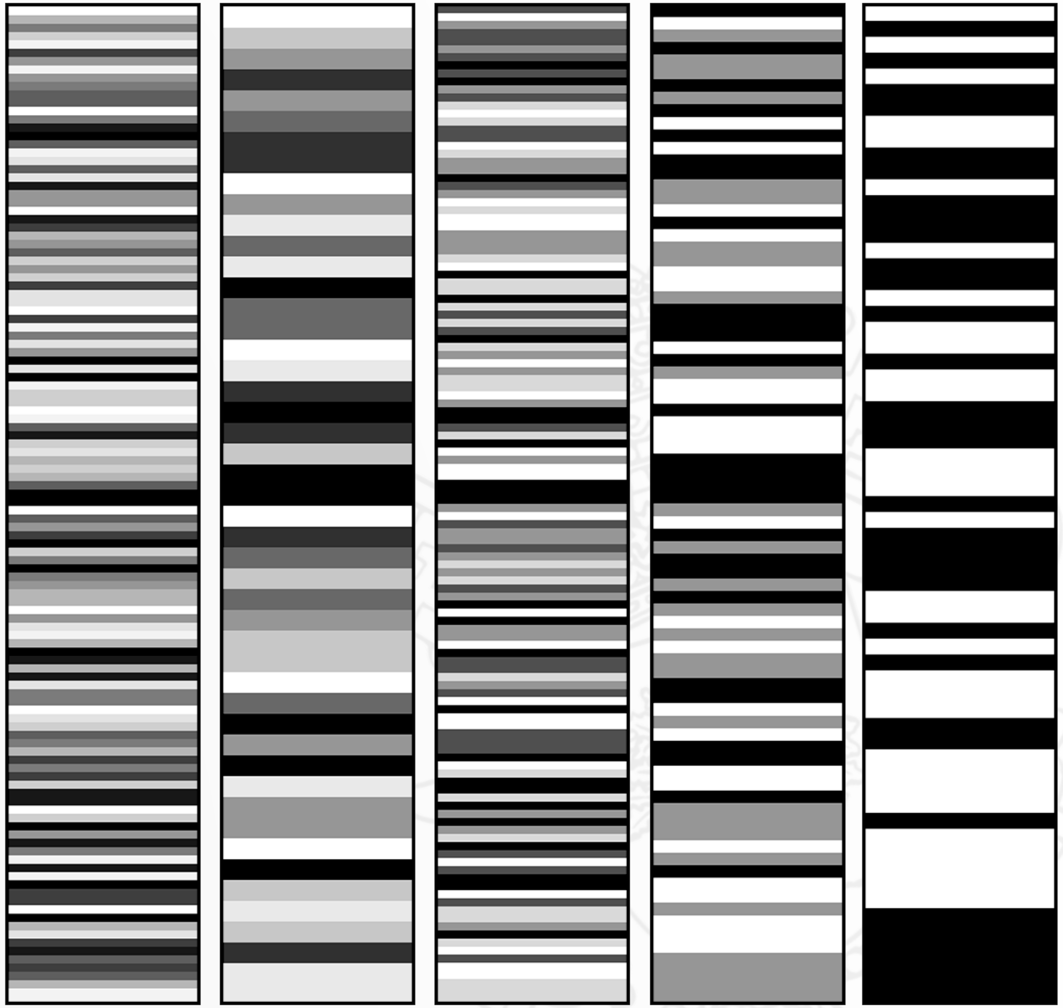
Universidad de Valladolid

Non-binary codes for c-VEP-based BCI systems

Víctor Martínez-Cagigal

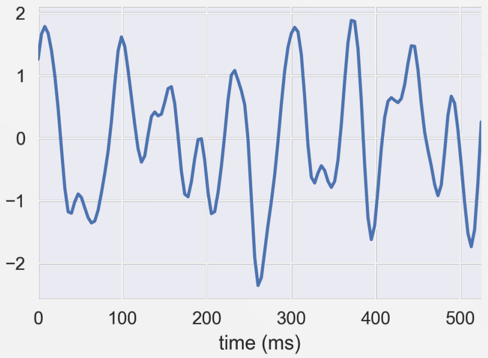
victor.martinez.cagigal@uva.es
www.gib.tel.uva.es
www.medusabci.com

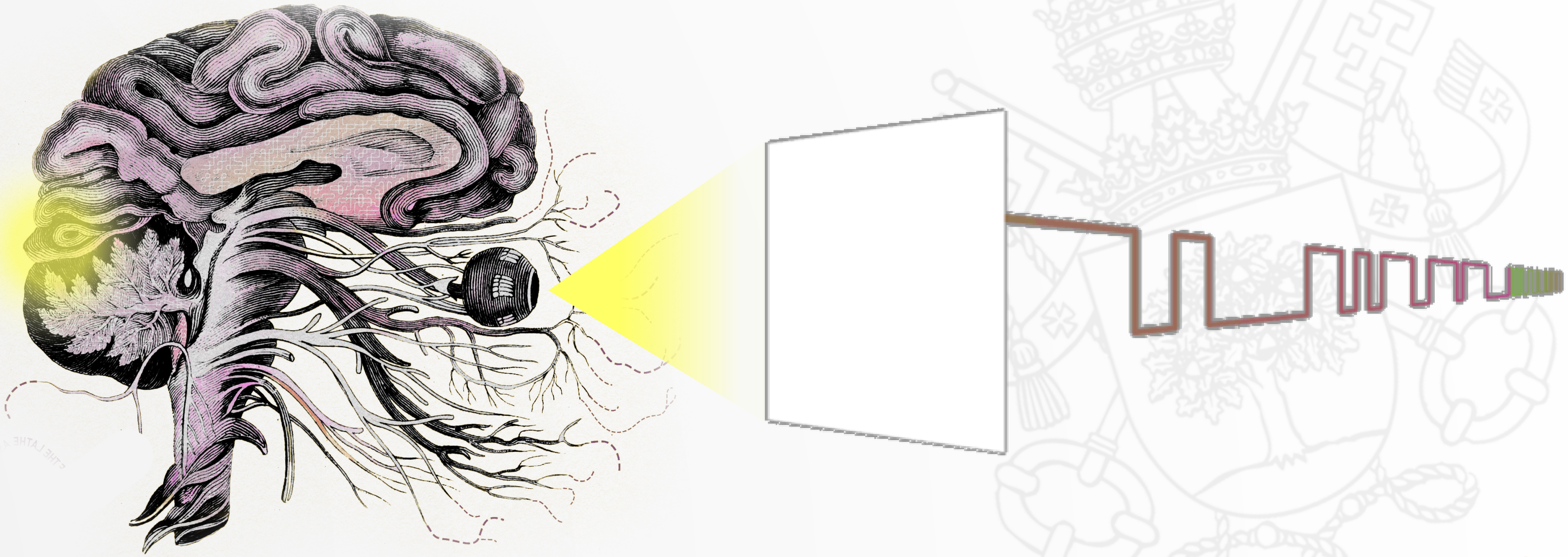
Contents



Introduction

- **Code-modulated visual evoked potentials (c-VEP)**
  - Visual evoked potentials in response to pseudorandom codes
  - Binary codes are usually encoded with black & white flashes
    - Maximal amplitude depth → stronger response at occipital cortex





Introduction

Cons

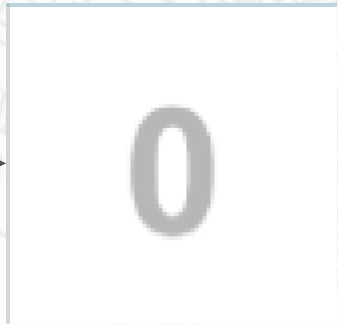
Pros

Eyestrain for some users

High accuracy (>90%)

Brief calibration (~30s)

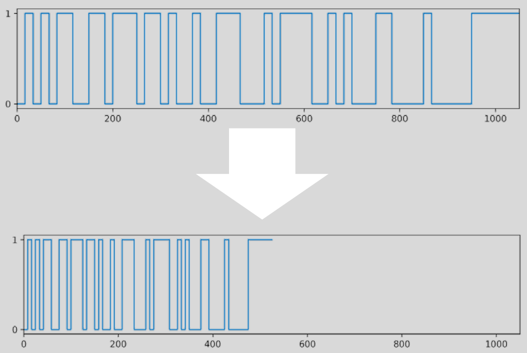
Visual fatigue may be caused by high-contrast stimuli!



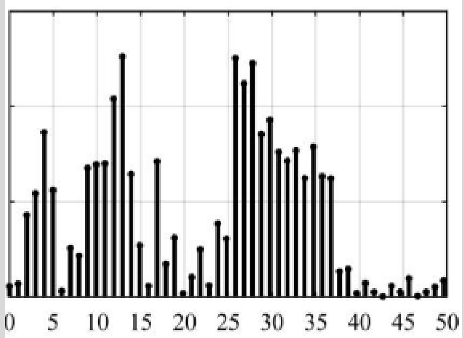
Introduction

- Addressing user convenience in c-VEP-based BCIs

1 Higher stimulation rates



2 High-frequency codes

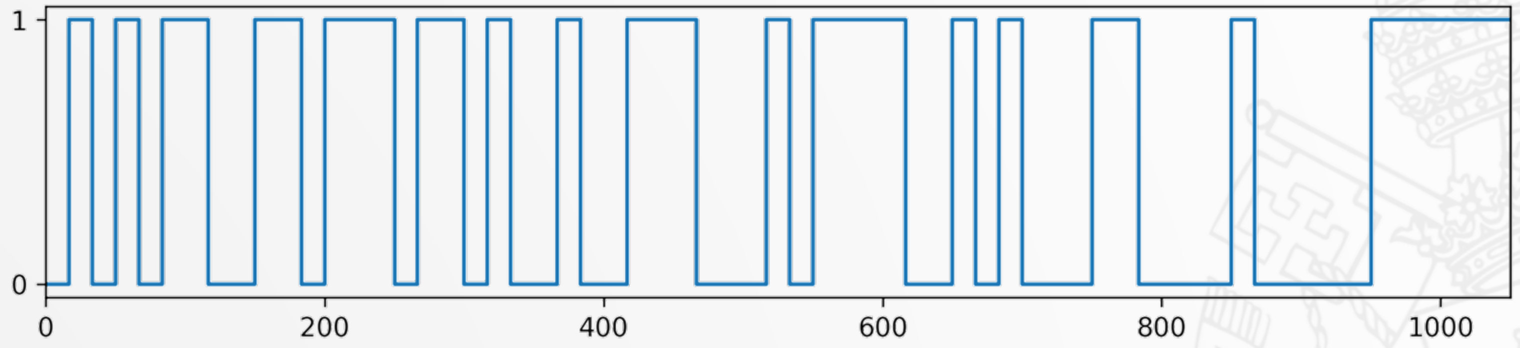


3 Non-binary codes

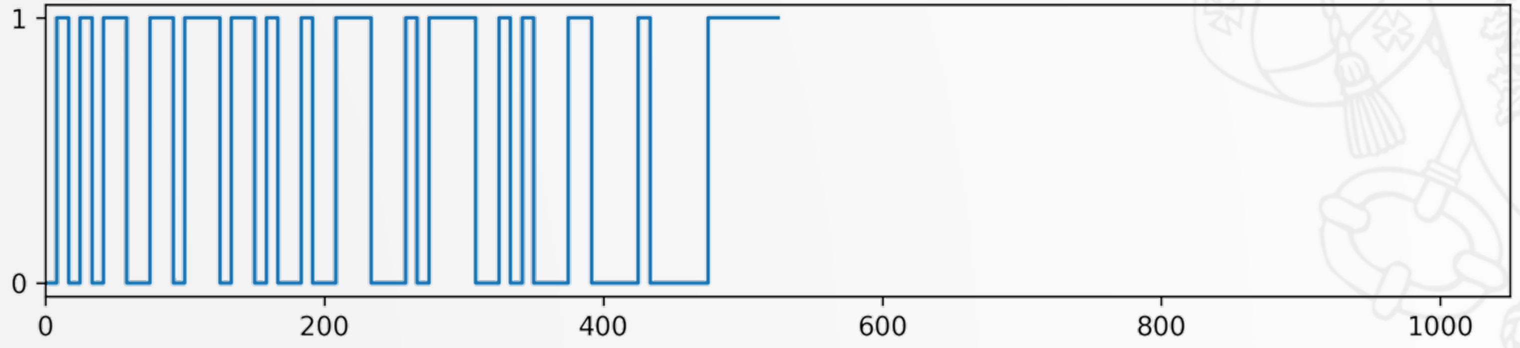
Introduction

- **Addressing user convenience in c-VEP-based BCIs**
  1. Higher stimulation rates
     - Many studies claimed that 120 Hz stimulation is less fatiguing than 60 Hz → 📖 Martínez-Cagigal et al., 2023
     - Although decreases in performance reported for >120 Hz rates (e.g. 240 Hz) → 📖 Gembler et al., 2018

60 Hz

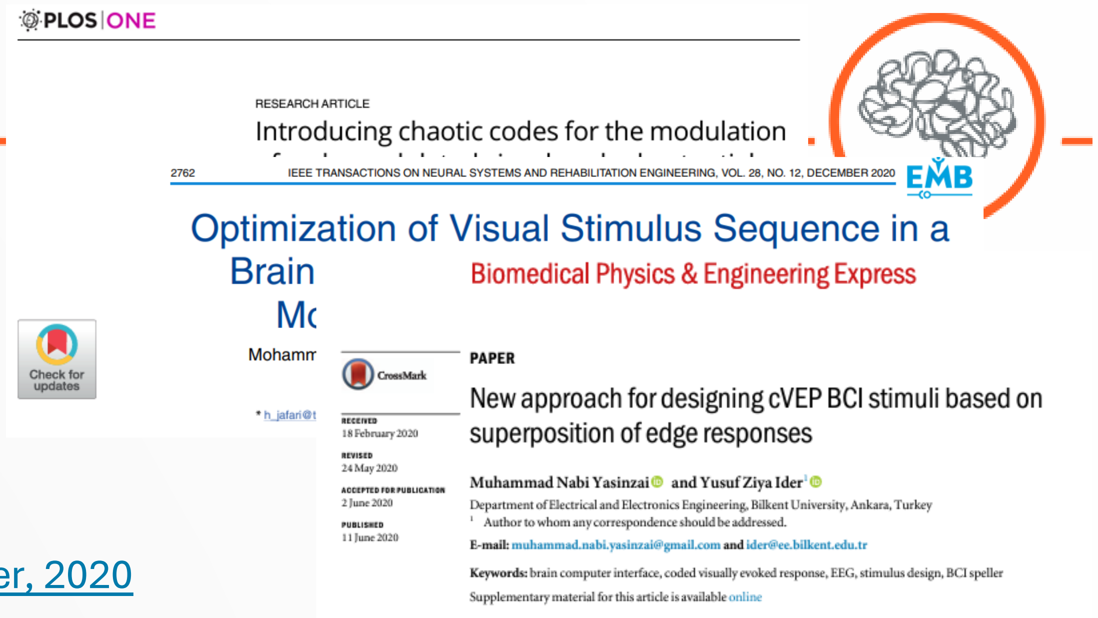


120 Hz



Introduction

- **Addressing user convenience in c-VEP-based BCIs**
    2. High-frequency codes
        - Chaotic codes → 📖 Shirzhiyan et al., 2019
        - Optimum sequences → 📖 Behboodi et al., 2020
        - Superposition optimized pulses → 📖 Yasinzai & Ider, 2020


PLOS ONE

RESEARCH ARTICLE

Introducing chaotic codes for the modulation

Check for updates

2762 IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING, VOL. 28, NO. 12, DECEMBER 2020

Optimization of Visual Stimulus Sequence in a Brain Mc

Mohamm

* h_jafari@t

Biomedical Physics & Engineering Express

CrossMark

RECEIVED 18 February 2020

REVISED 24 May 2020

ACCEPTED FOR PUBLICATION 2 June 2020

PUBLISHED 11 June 2020

PAPER

New approach for designing cVEP BCI stimuli based on superposition of edge responses

Muhammad Nabi Yasinzai and Yusuf Ziya Ider[1]

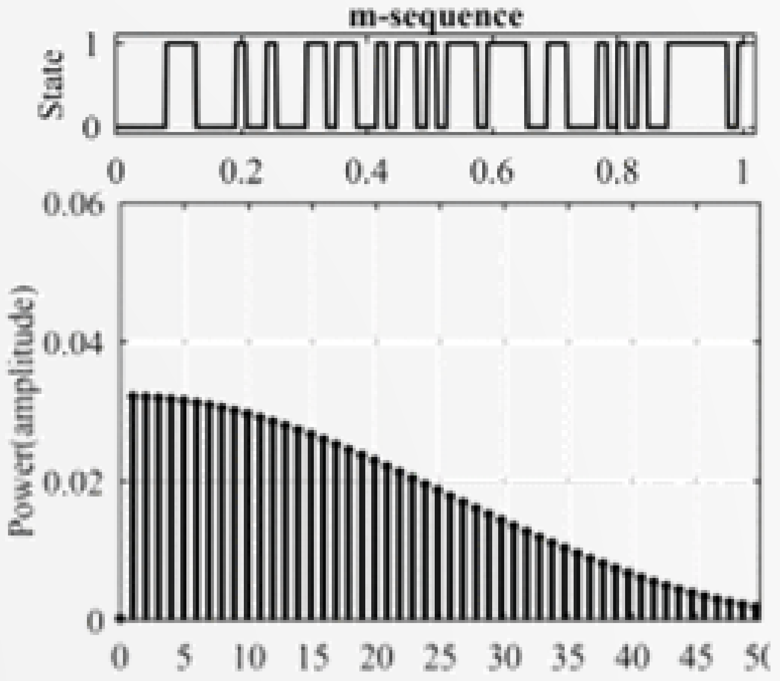
Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey

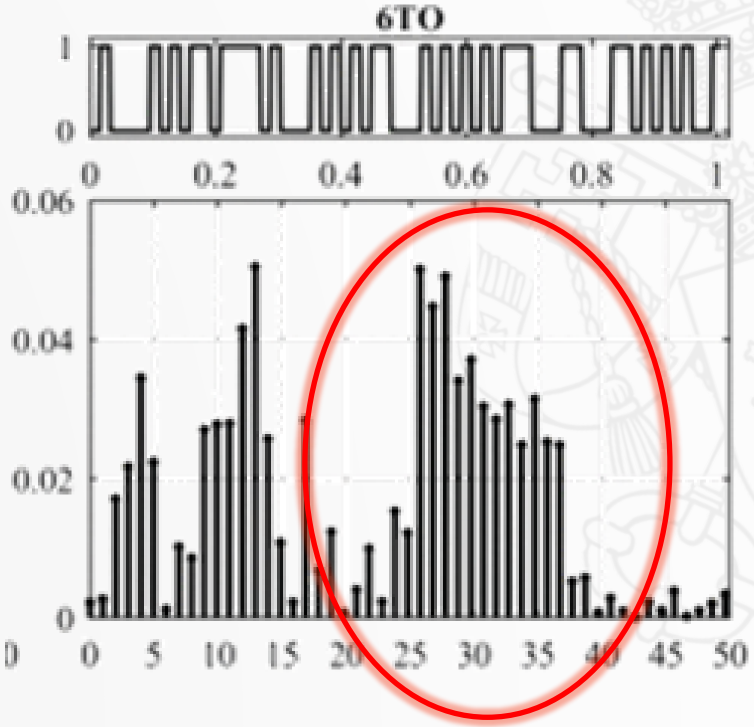
[1] Author to whom any correspondence should be addressed.

E-mail: muhammad.nabi.yasinzai@gmail.com and ider@ee.bilkent.edu.tr

Keywords: brain computer interface, coded visually evoked response, EEG, stimulus design, BCI speller

Supplementary material for this article is available online

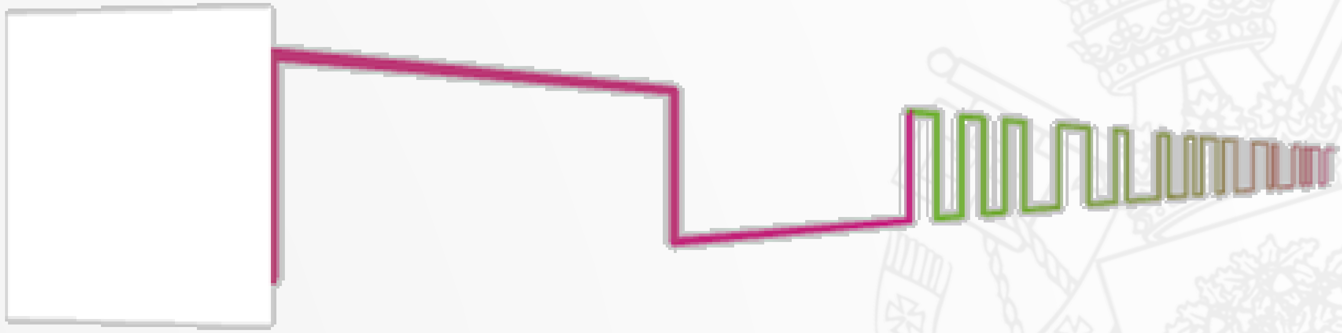




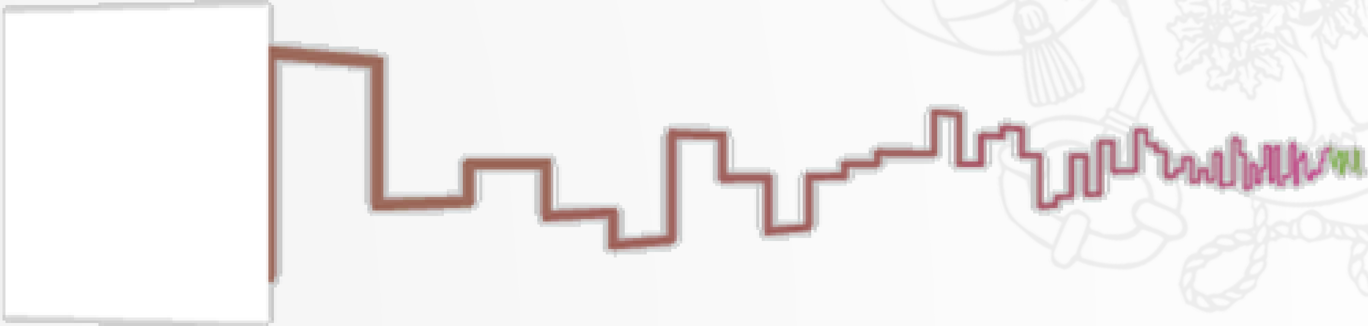
Introduction

- **Addressing user convenience in c-VEP-based BCIs**
  3. Non-binary codes
     - I.e., codes with more levels than just 0 and 1
     - We can encode them with different shades of gray and colors to reduce amplitude depth
     - Demonstrated to be more comfortable than binary codes → Martínez-Cagigal et al., 2023

Binary (base 2)

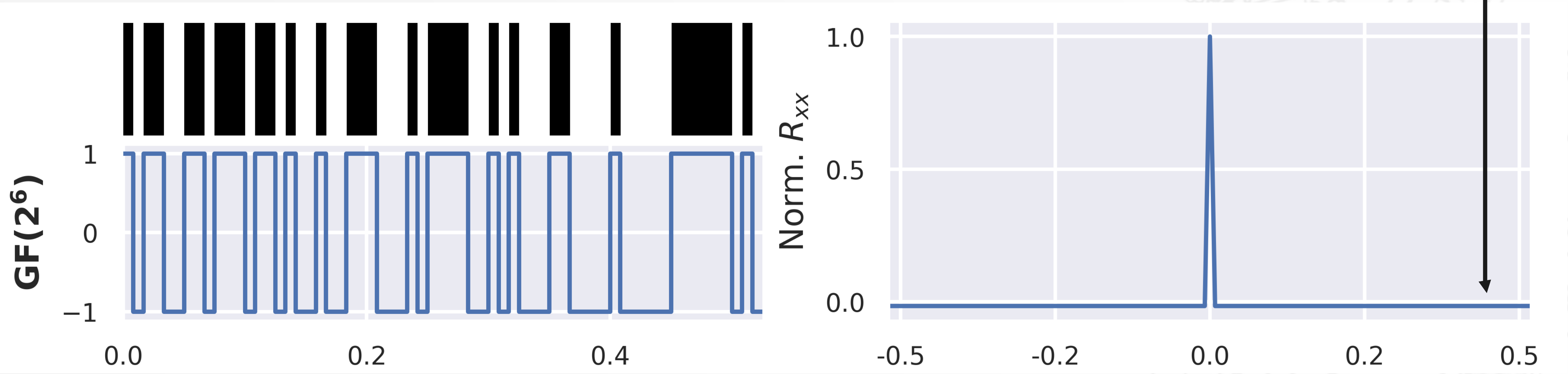


11-ary (base 11)



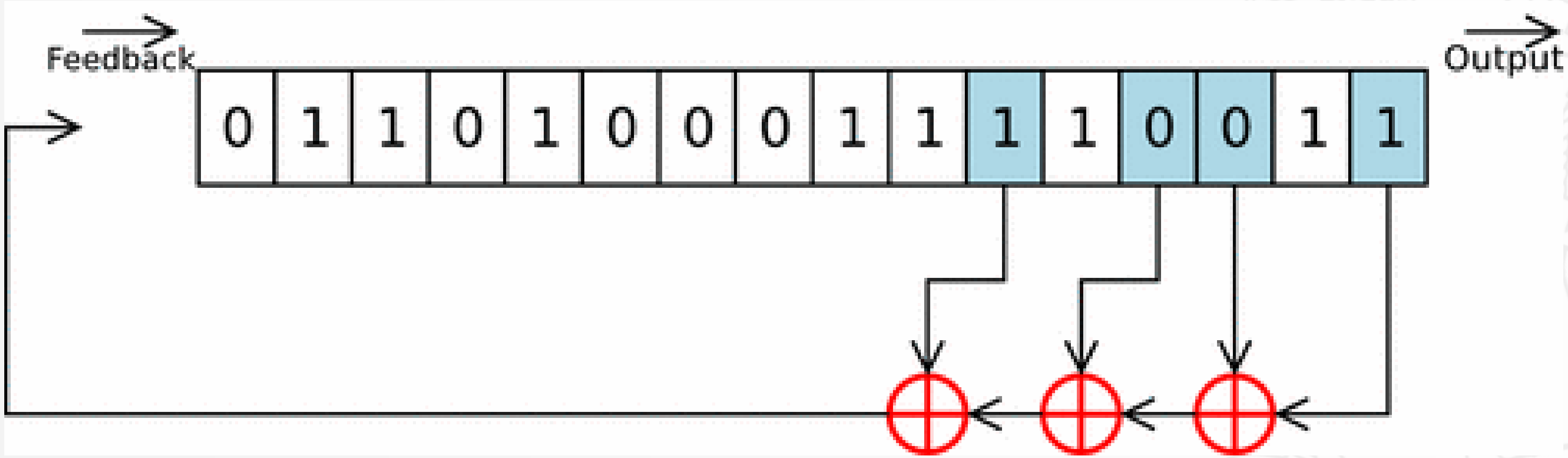
Maximal-length Sequences

- **What are maximal-length sequences (m-sequences)?**
  - Pseudorandom periodic time series
  - With optimal autocorrelation profiles
    - For binary codes, -1/N for any shift different than 0



Maximal-length Sequences

- **Linear Feedback Shift Registers (LFSR)**
  - Shift registers that compute new values using a linear function of the immediate previous state
  - The code is generated by appending the output to the initial position of the state



Maximal-Length Sequences

- **Generation of *p*-ary m-sequences**
  - The length of the output code depends on:

$$L = p^r - 1$$

Length

Base

Polynomial order

  - Notation:

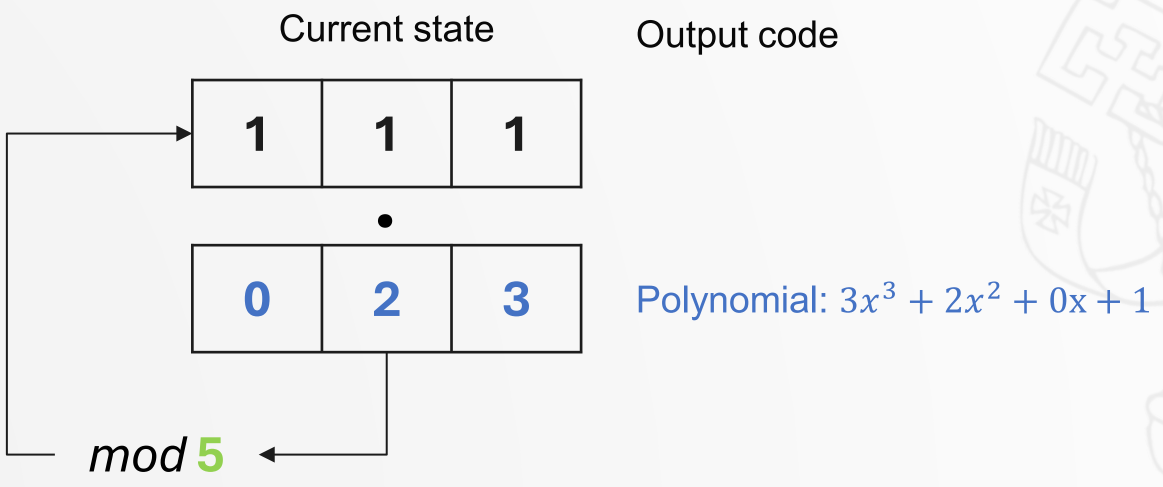
$$\mathrm{GF}(p^r) \longrightarrow \text{For } \mathrm{GF}(2^6),\ L = 63 \text{ bits}$$

  - Requirement: using primitive polynomials over Galois Field $\mathrm{GF}(p^r)$
    1. Only bases where p is prime are allowed (e.g., 2, 3, 5, 7, 11...)
    2. The codes are fixed in length (i.e., $L = p^r - 1$)

Maximal-Length Sequences

- **Generation of *p*-ary m-sequences**
  - Each bit is calculated by:
    1. Applying matrix product between the current state and the polynomial
    2. Then applying the modulo p operation
  - Example for base 5 → $\mathrm{GF}(5^3)$

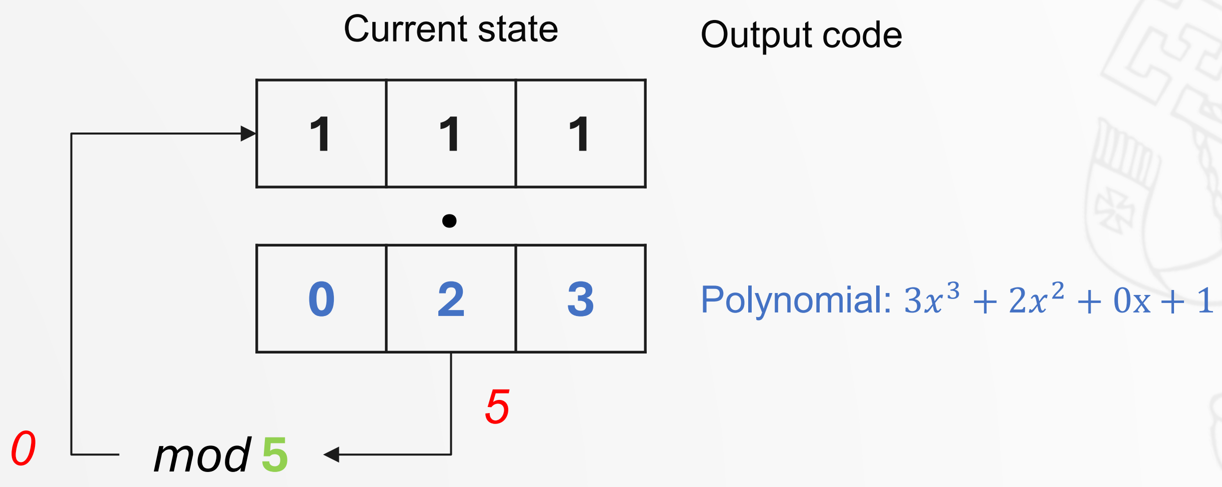
Base	Order	Length	Polynomial
5	3	124 bits	$3x^3 + 2x^2 + 1$



Maximal-Length Sequences

- **Generation of *p*-ary m-sequences**
  - Each bit is calculated by:
    1. Applying matrix product between the current state and the polynomial
    2. Then applying the modulo p operation
  - Example for base 5 → $\mathrm{GF}(5^3)$

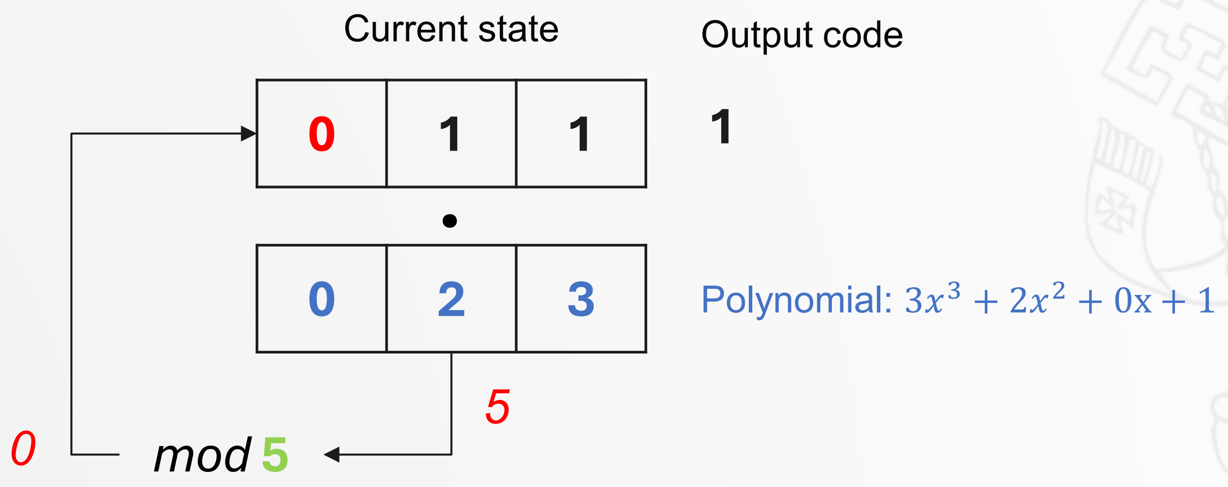
Base	Order	Length	Polynomial
5	3	124 bits	$3x^3 + 2x^2 + 1$



Maximal-Length Sequences

- **Generation of *p*-ary m-sequences**
  - Each bit is calculated by:
    1. Applying matrix product between the current state and the polynomial
    2. Then applying the modulo p operation
  - Example for base 5 → $\mathrm{GF}(5^3)$

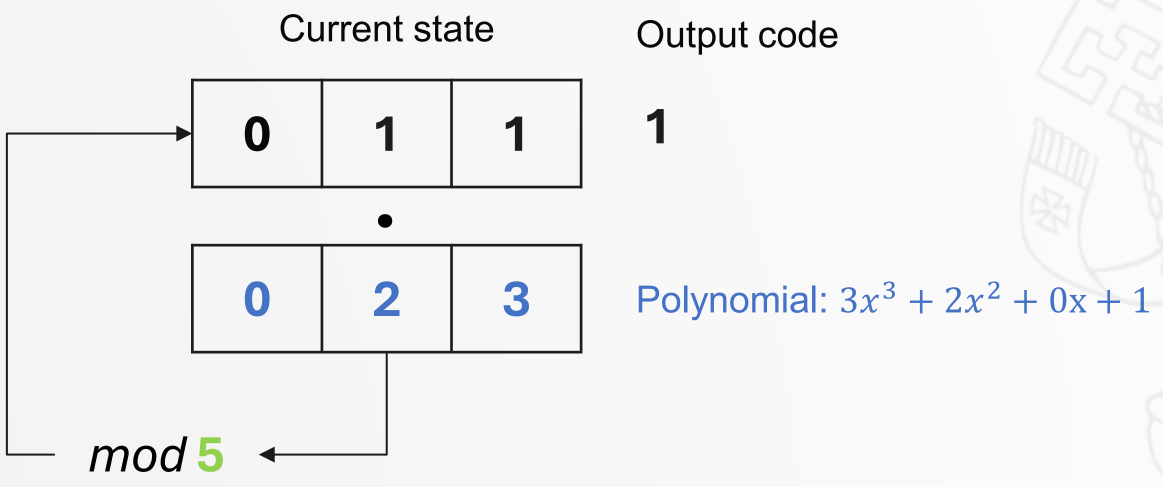
Base	Order	Length	Polynomial
5	3	124 bits	$3x^3 + 2x^2 + 1$



Maximal-Length Sequences

- **Generation of *p*-ary m-sequences**
  - Each bit is calculated by:
    1. Applying matrix product between the current state and the polynomial
    2. Then applying the modulo p operation
  - Example for base 5 → $\mathrm{GF}(5^3)$

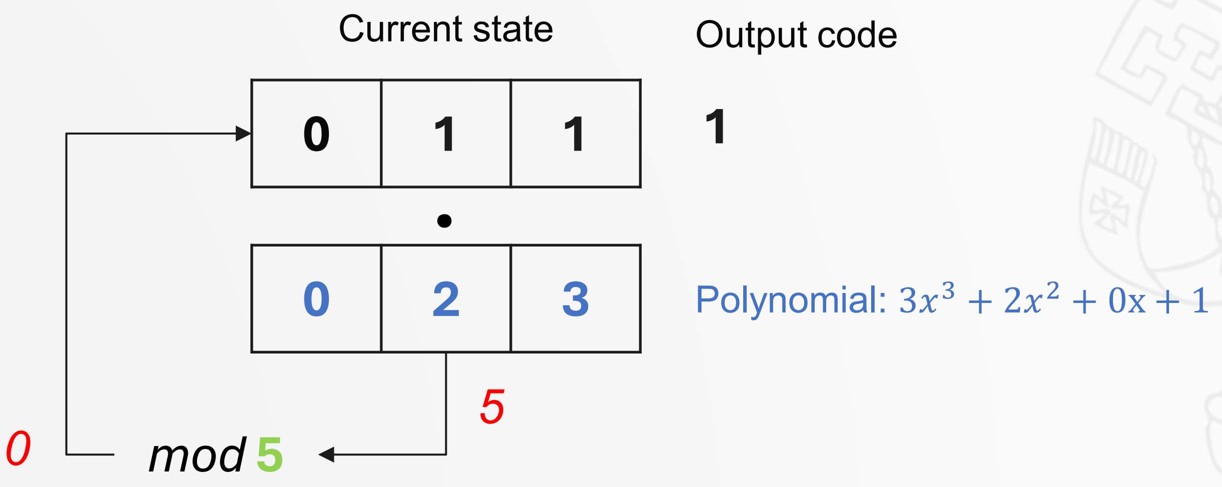
Base	Order	Length	Polynomial
5	3	124 bits	$3x^3 + 2x^2 + 1$



Maximal-Length Sequences

- **Generation of *p*-ary m-sequences**
  - Each bit is calculated by:
    1. Applying matrix product between the current state and the polynomial
    2. Then applying the modulo p operation
  - Example for base 5 → $\mathrm{GF}(5^3)$

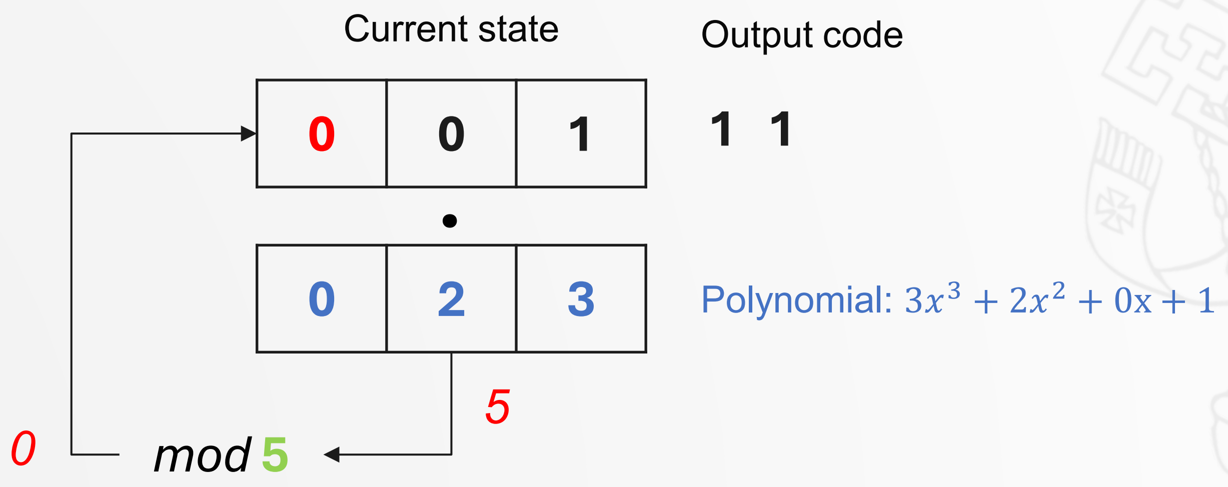
Base	Order	Length	Polynomial
5	3	124 bits	$3x^3 + 2x^2 + 1$



Maximal-Length Sequences

- **Generation of *p*-ary m-sequences**
  - Each bit is calculated by:
    1. Applying matrix product between the current state and the polynomial
    2. Then applying the modulo p operation
  - Example for base 5 → $\mathrm{GF}(5^3)$

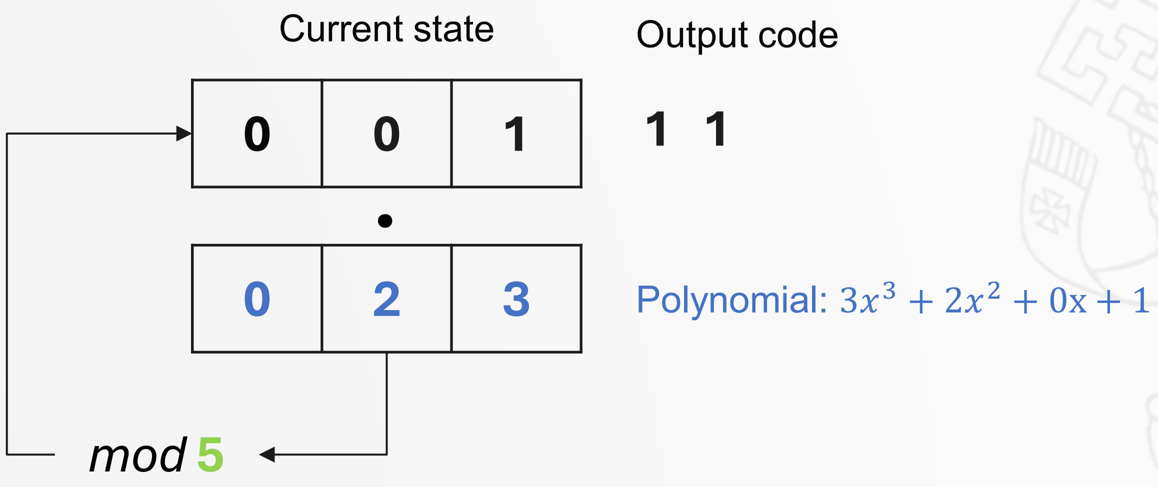
Base	Order	Length	Polynomial
5	3	124 bits	$3x^3 + 2x^2 + 1$



Maximal-Length Sequences

- **Generation of *p*-ary m-sequences**
  - Each bit is calculated by:
    1. Applying matrix product between the current state and the polynomial
    2. Then applying the modulo p operation
  - Example for base 5 → $\mathrm{GF}(5^3)$

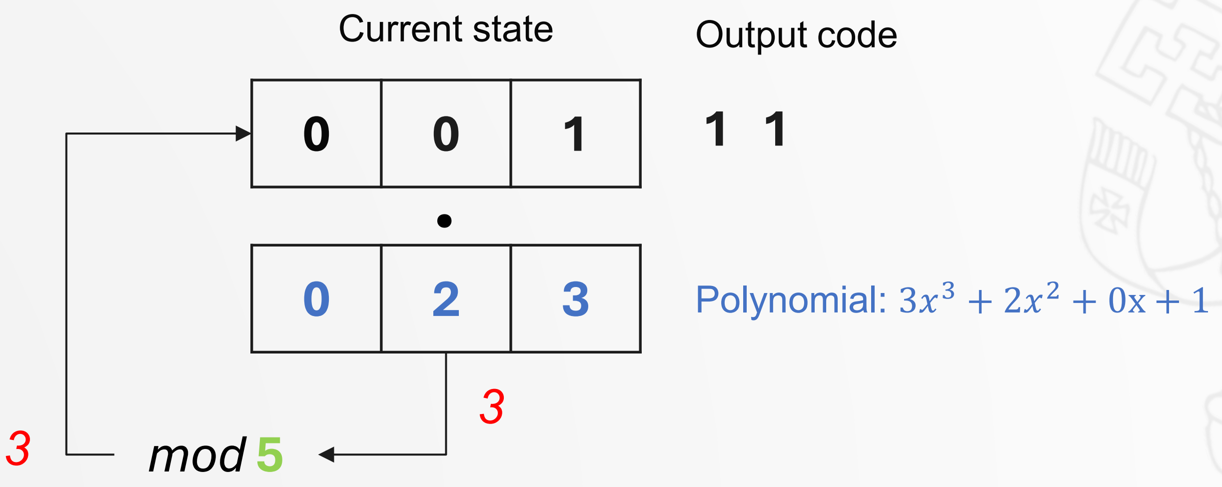
Base	Order	Length	Polynomial
5	3	124 bits	$3x^3 + 2x^2 + 1$



Maximal-Length Sequences

- **Generation of *p*-ary m-sequences**
  - Each bit is calculated by:
    1. Applying matrix product between the current state and the polynomial
    2. Then applying the modulo p operation
  - Example for base 5 → $\mathrm{GF}(5^3)$

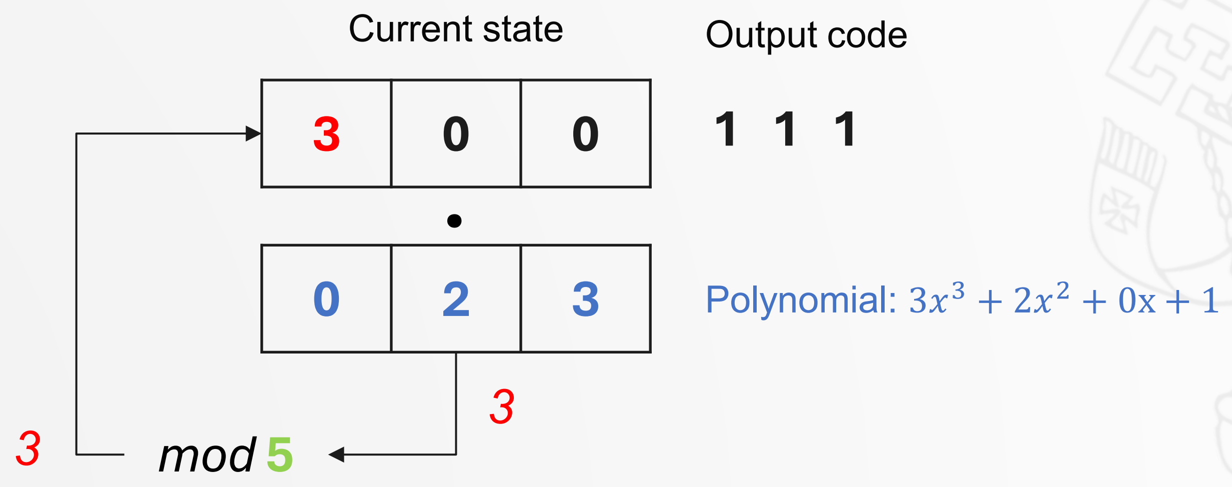
Base	Order	Length	Polynomial
5	3	124 bits	$3x^3 + 2x^2 + 1$



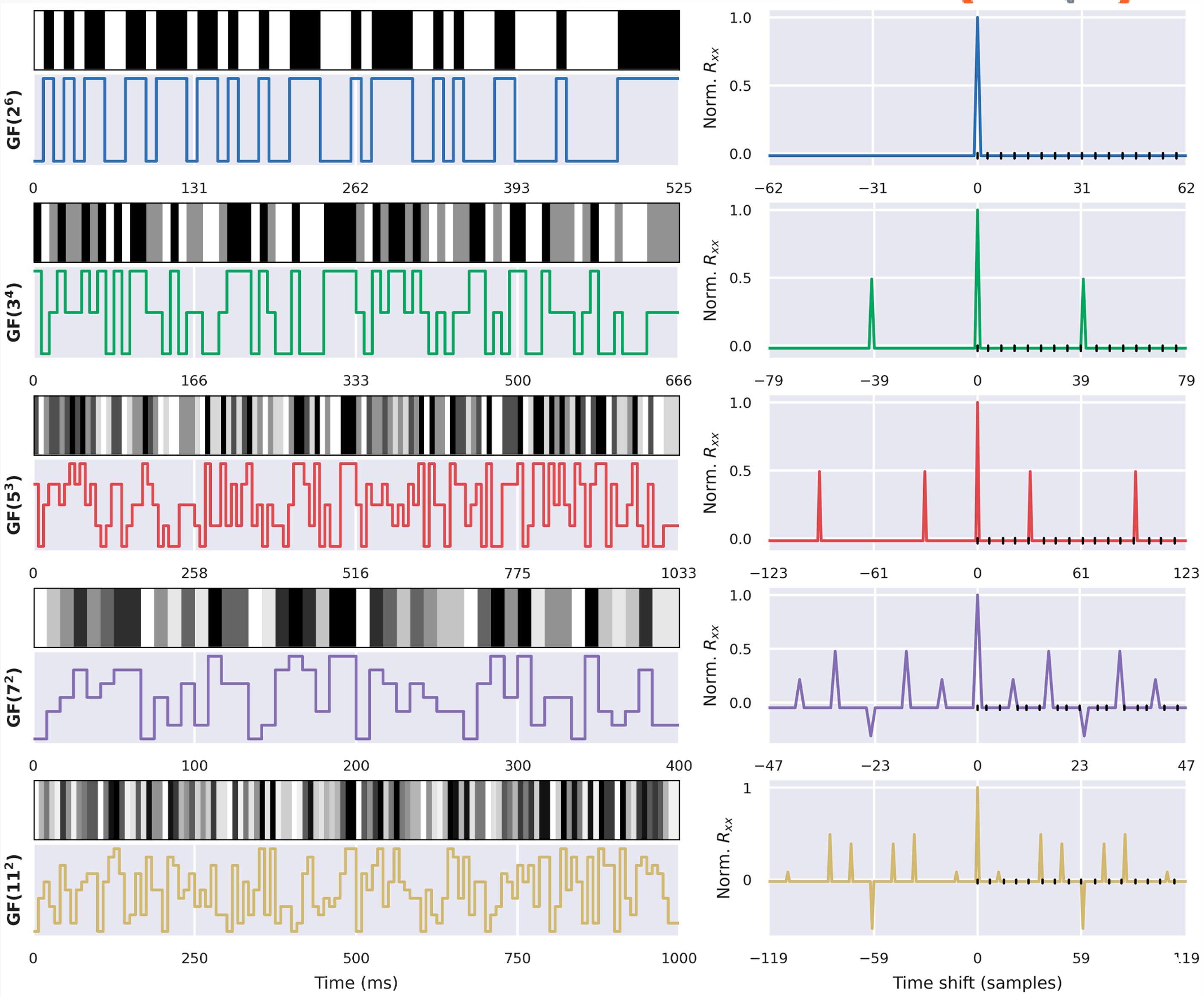
Maximal-Length Sequences

- **Generation of *p*-ary m-sequences**
  - Each bit is calculated by:
    1. Applying matrix product between the current state and the polynomial
    2. Then applying the modulo p operation
  - Example for base 5 → $\mathrm{GF}(5^3)$

Base	Order	Length	Polynomial
5	3	124 bits	$3x^3 + 2x^2 + 1$

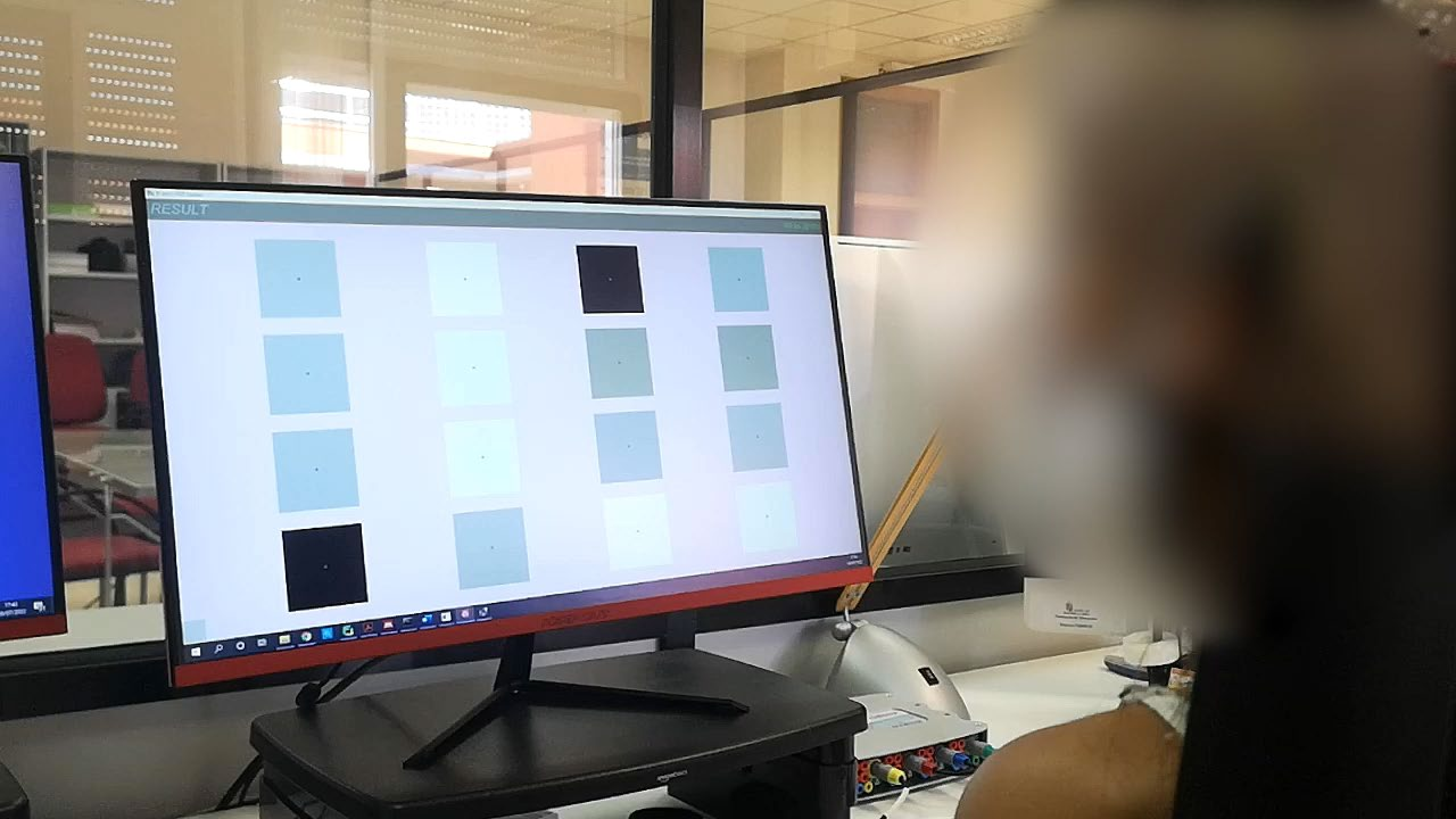


Correlation local maxima



Generation details of the p-ary m-sequences.

	Base	Order	Length (bits)	Polynomial	Duration (s/cycle) 120 Hz	Duration (s/cycle) 60 Hz
GF(2^6)	2	6	63	$x^6 + x^5 + 1$	0.525	1.050
GF(3^4)	3	4	80	$x^4 + 2x^3 + 1$	0.667	1.333
GF(5^3)	5	3	124	$3x^3 + 2x^2 + 1$	1.033	2.067
GF(7^2)	7	2	48	$4x^2 + 1$	0.408	0.800
GF(11^2)	11	2	120	$3x^2 + x + 1$	1.000	2.000


RESULT

Evidence of benefits of non-binary codes

Expert Systems With Applications 232 (2023) 120815

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

ELSEVIER

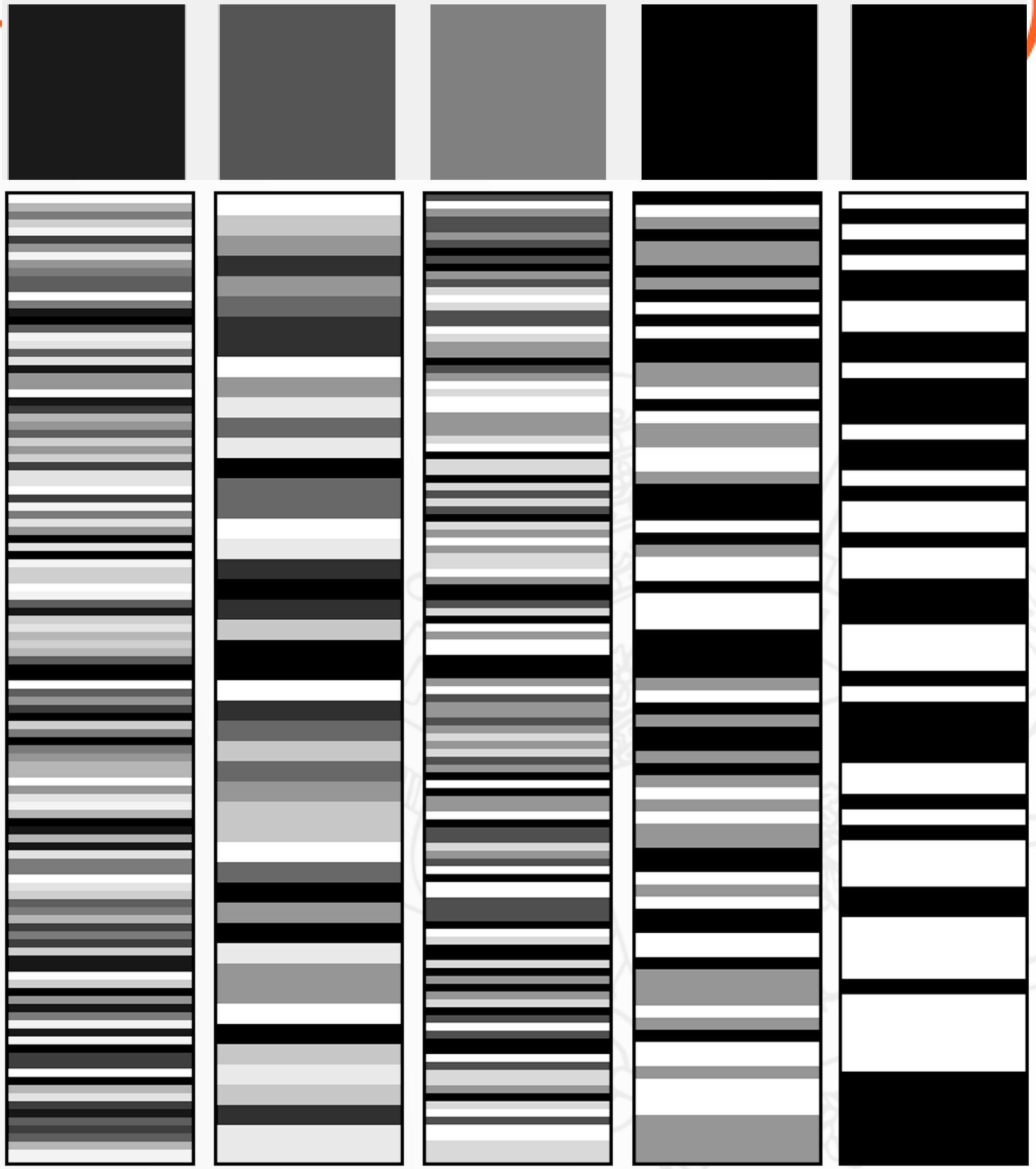
Non-binary m-sequences for more comfortable brain–computer interfaces based on c-VEPs

Víctor Martínez-Cagigal [a,b,*], Eduardo Santamaría-Vázquez [a,b], Sergio Pérez-Velasco [a,b], Diego Marcos-Martínez [a,b], Selene Moreno-Calderón [a], Roberto Hornero [a,b]

SCAN ME

Evidence of benefits of non-binary codes

- **Our study**
  - Five different *p*-ary m-sequences (bases 2, 3, 5, 7 and 11)
  - Reference signal processing (📖 Martínez-Cagigal et al., 2021)
  - Equipment
    - g.USBamp with 16 active channels
    - MEDUSA© software (www.medusabci.com)
  - Experimental protocol
    - 15 healthy subjects (28.80 ± 5.02 years)
    - Calibration: 300 cycles for each sequence
    - Online: 32 trials (10 cycles/trial)
    - Command matrix of 4×4 dimensions



Evidence of benefits of non-binary codes

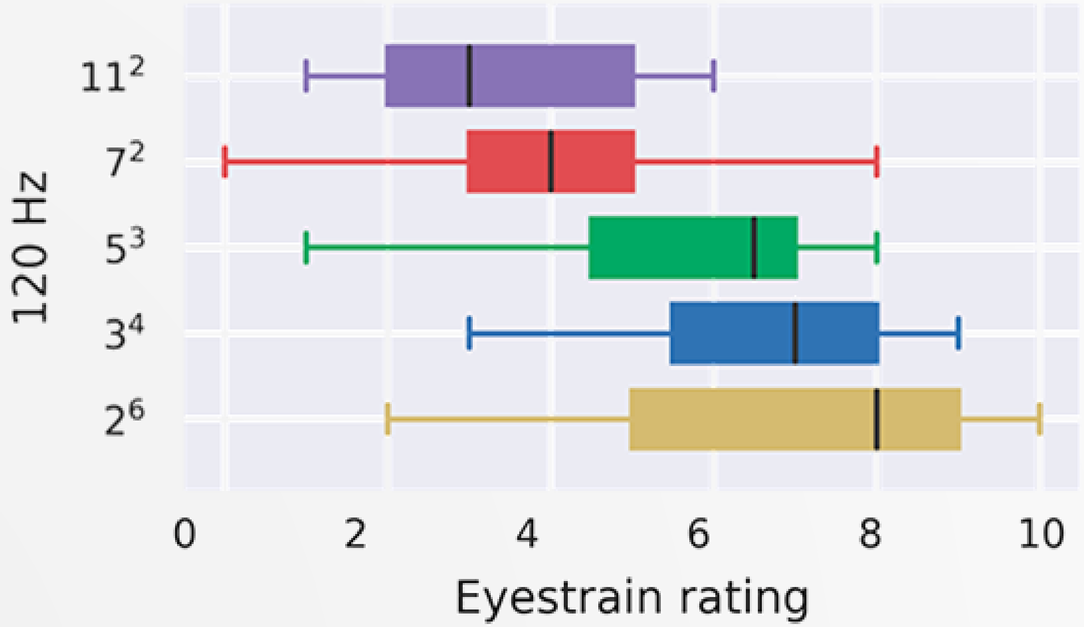
- **Performance analysis**
  - Users were able to reach more than 98% grand-averaged accuracy with 10 cycles using all the proposed p-ary m-sequences

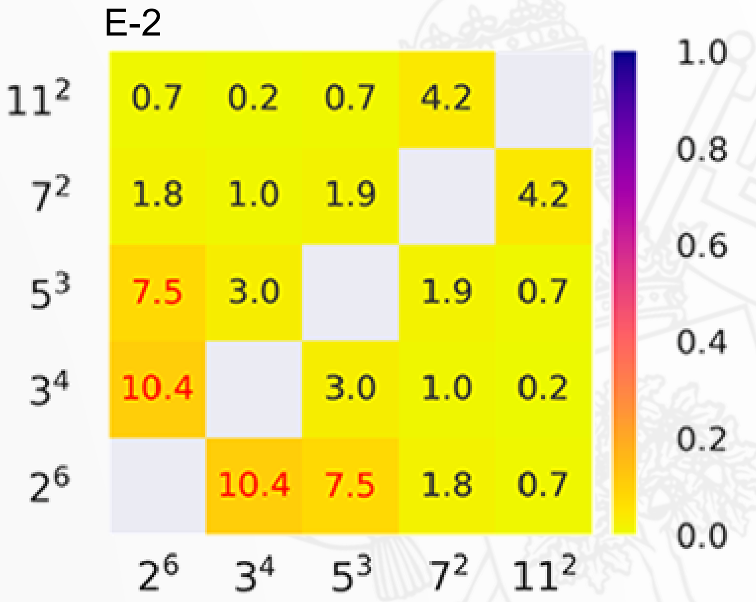
Grand-averaged online results across users.

	No. cycles	1	2	3	4	5	6	7	8	9	10	Mean ± STD
GF(2^6)	Accuracy (%)	65.43	90.43	96.68	98.05	**99.41**	**99.41**	**99.61**	**99.61**	**99.41**	**99.61**	94.77 ± 10.14
	ITR (bpm)	208.06	188.03	141.04	108.91	90.14	75.11	64.65	56.57	50.08	45.25	102.78 ± 55.19
	Duration (s)	0.53	1.05	1.58	2.10	2.62	3.15	3.68	4.20	4.73	5.25	2.89 ± 1.51
GF(3^4)	Accuracy (%)	75.78	92.77	97.27	97.66	98.44	98.83	99.02	99.02	99.02	99.22	95.70 ± 6.89
	ITR (bpm)	216.59	154.70	113.27	85.68	69.67	58.54	50.31	44.02	39.13	35.36	86.73 ± 55.97
	Duration (s)	0.67	1.33	2.00	2.67	3.33	4.00	4.67	5.33	6.00	6.67	3.67 ± 1.91
GF(5^3)	Accuracy (%)	83.20	94.14	**98.05**	**98.83**	99.02	99.22	99.41	99.41	99.22	99.22	96.97 ± 4.83
	ITR (bpm)	163.72	103.05	74.01	56.54	45.47	38.02	32.76	28.66	25.39	22.85	59.05 ± 42.20
	Duration (s)	1.03	2.07	3.10	4.13	5.17	6.20	7.23	8.27	9.30	10.33	5.68 ± 2.97
GF(7^2)	Accuracy (%)	49.80	82.23	93.36	95.90	97.27	97.46	98.05	97.66	98.24	98.05	90.80 ± 14.42
	ITR (bpm)	168.14	205.30	174.38	137.45	113.43	95.18	82.46	71.65	64.31	57.72	117.00 ± 49.15
	Duration (s)	0.40	0.80	1.20	1.60	2.00	2.40	2.80	3.20	3.60	4.00	2.20 ± 1.15
GF(11^2)	Accuracy (%)	**86.72**	**97.46**	**98.05**	98.63	98.83	98.63	98.63	98.63	98.63	98.44	**97.27** ± 3.54
	ITR (bpm)	179.64	113.79	76.79	58.36	46.93	38.99	33.42	29.24	25.99	23.33	62.65 ± 47.10
	Duration (s)	1.00	2.00	3.00	4.00	5.00	6.00	7.00	8.00	9.00	10.00	5.50 ± 2.87

Evidence of benefits of non-binary codes

- **Eyestrain analysis**
  - The higher the base (i.e., number of levels), the less eyestrain users perceive
    - E.g., GF(7^2) and GF(11^2) are significantly less fatiguing than binary GF(2^6)

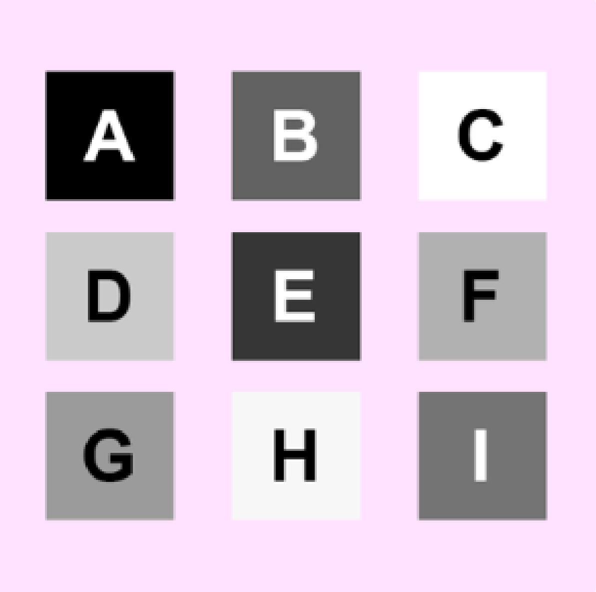




Black: significant differences (p-value < 0.05)
Red: no differences (p-value > 0.05)

Open-source software for non-binary codes

www.medusabci.com



vicmarcag

P-ary c-VEP Speller

A circular-shifting c-VEP speller that supports non-binary m-sequences

☆ 0.0 37

Any questions?

9th Graz Brain-Computer Interface Conference

Join Forces - Increase Performance

GRAZ BCI

BIOMEDICAL ENGINEERING GROUP
UNIVERSITY OF VALLADOLID

Universidad de Valladolid

Non-binary codes for c-VEP-based BCI systems

Víctor Martínez-Cagigal

victor.martinez.cagigal@uva.es

www.gib.tel.uva.es

www.medusabci.com