Chapter 2

Error-detecting codes based on T-quasigroup

In modern communication systems, the most frequent errors that occur during data transmission are system errors, sampling errors, measurement errors, data tampering, data corruption, and burst errors. These errors can significantly impact the accuracy and integrity of the transmitted data, and therefore it is essential to detect and correct them. Verhoeff and Beckley, in their statistical studies [6, 124], concluded that human operators during data entry or data transmission over noisy channels are most prone to the errors listed in Table 2.1. Intentionally, in Table 2.1 we exclude the errors like insertion and deletion of redundant bits because we assume that if all the codewords are of equal length then this kind of errors can be easily detected.

Types	Error type		Relative frequency in %	
			Verhoeff	Beckley
1.	Single error	$a \to b$	$79.0(60-95)$	86
2.	adjacent error	$ab \to ba$	10.2	8
3.	jump transposition	$abc \to cba$	0.8	8
4.	twin error	$aa \to bb$	0.6	6
5.	phonetic error ($a \geq 2$)	$a0 \to 1a$	0.5	6
6.	jump twin error	$aca \to bcb$	0.3	6
7.	other error		8.6	6

Table 2.1: Error types and their frequencies

Researchers worldwide strive to develop error detection systems that are effective against all error types, prioritizing those with higher frequencies. There are various techniques for error detection like parity check, checksum and cyclic

redundancy check.

In literature various error-detecting codes have been proposed based on associative structure like groups and finite fields. In 1969, Verhoeff [124] proposed a decimal code over the Dihedral group D_5 by utilizing an appropriate permutation. This code can detect double errors of type (1) and type (2) mentioned in Table 2.1. In 1985, Gumm [66] proposed a general check digit system over a group of order $r = 2s$ where s is an odd prime. He utilized the dihedral group D_r with an appropriate permutation. Similar to Verhoeff System, this system can also detect errors of type (1) and type (2). In 1997, Broecker [16] proposed a check digit system based on Chevalley groups with even characteristic, capable of detecting errors of types (1) – (5). In 2011, Niemenmaa [98], proposed a check digit system for hexadecimal numbers that utilizes an automorphism of an Abelian group having order 16, instead of a permutation like Verhoeff system [124]. This system has the capability to detect all error types (1) – (5) as defined in Table 2.1. Notably, it is included in a video community's *MISB standard 1204.1*. In 2012, Chen et al. [26] generalized Niemenmaa's work [98] by proposing a check digit system over an Abelian group having order p^k, for p-prime and $k \geq 1$. However, both systems offer same error detection capabilities.

In 2004, Damm [30] constructed a decimal code over totally anti-symmetric quasigroup of order 10. This decimal code is more efficient than the Verhoeff decimal code [124] because it is built without using any specific permutation. In same year, Mullen and Shcherbacov [97] proposed a check digit system based on n-T-quasigroup which has the capability to detect errors of types (1)–(5). Similarly, in 2005, Belyavskaya et al. [12, 13] proposed several check character system using quasigroups .

Moreover, the research community is actively working on designing codes using quasigroups which can detect and correct errors simultaneously. There are many well-known real life practical examples in which check digit systems are used:

- The International Standard Book Number (ISBN) code uniquely identifies a specific edition of a book. Before 2007, ISBN-10 was used. However, to increase global compatibility, ISBN-13 was introduced. This 13-digit version ISBN number aligns with the European Article Number (EAN)-code.

- Social Security Number (SSN) is a unique identifier issued by United states of America. It serves various purposes, including identification of a person and taxation. To detect the errors in these types of code is crucial to prevent frauds and online illegal activity.

- Similar to the SSN, Aadhaar is a unique 12-digit identification number issued

by the Indian government to all residents. It links an individual's biometric and demographic data. Aadhaar generation involves quality check, packet validation, demographic and de-duplication of biometric to ensure uniqueness. For verification, Aadhaar utilizes a checksum digit formula described by Verhoeff algorithm [124]. This algorithm generates a final digit that verifies the validity of the first eleven digits.

The aforementioned check digit systems can detect (or correct) single and some double errors in an erroneous codeword. However, they are not capable of detecting burst errors, where multiple bits in a row are corrupted. The main purposes of this chapter is to propose a check block system capable of detecting the different types of burst errors listed in Table 2.1 as a block. In a check block system, the check block is generally appended to the codeword and identify errors in an erroneous codeword.

We organize this chapter into several sections. In Section 2.1, we discuss the definition of general check digit system and some results on finite fields used later in the chapter. In Section 2.2, we propose a check block system based on T-quasigroup utilizing field automorphisms of $\mathbb{F}_{p^n}/\mathbb{F}_p$. Additionally, we give the necessary and sufficient conditions for the detection of single errors, k- jump transposition errors, k- jump twin errors, and phonetic errors in an erroneous codeword blocks. In Section 2.3, we propose a check digit system based on the T-quasigroup, employing group automorphism mapping over $(\mathbb{F}_p, +)$, with necessary and sufficient conditions for the detection of single and double errors such as transposition errors, twin errors, and phonetic errors. In Section 2.4, we analyze the advantage of our check block system over Reed-Solomon code focusing on number of operations require to detect single position errors. We prove that our code can be seamlessly integrated into real-life application like ISBN code over group $(\mathbb{F}_p, +)$, SSN and Bank routing number over $\mathbb{F}_{p^n}$.

2.1 General check digit system

Definition 2.1.1. [116] Consider automorphisms $\alpha_1, \alpha_2, \ldots, \alpha_n$ of the group $(Q, +)$, a be a fixed element from the set Q and define an n-ary operation as:

$$g(x_1, \ldots, x_n) = \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n + a = \sum_{i=1}^{n} \alpha_i x_i + a. \tag{2.1}$$

Then, (Q, g) forms the linear n-ary quasigroup over the group $(Q, +)$.

Remark 2.1.2. A linear n-ary quasigroup over an Abelian group $(Q, +)$ is called

an n-T-quasigroup.

Definition 2.1.3. [116] A systematic error detecting code over an alphabet set Q can be established by appending a check digit, denoted by a_{n+1}, to each word $a_1a_2\ldots a_n \in Q^n$. Therefore, a systematic error-detecting code C, can be described as:

$$C : \begin{cases} Q^n \longrightarrow Q^{n+1} \\ a_1a_2\ldots a_n \longrightarrow a_1a_2\ldots a_na_{n+1} \end{cases} \tag{2.2}$$

Note: For code C, the check digit/character, which is an additional symbol used for error detection, is typically placed at the end of each codeword.

Definition 2.1.4. [116] Consider a code C with codewords having length $n+1$. Each codeword includes a single check character $x_{n+1} = g(x_1, \ldots, x_n)$ over an alphabet set Q is known to be an n-ary code, where g being a function from Q^n to Q. If, in an n-ary code (Q, g) the operation g is an n-ary quasigroup operation, then code C is an n-ary quasigroup code (Q, g).

If the $(n+1)^{th}$ character of the code C is a constant, we usually omit it and thus reducing it to length n. After defining the n-ary quasigroup code (Q, g). Now, we define an n-T quasigroup code which is defined over n-ary T quasigroup.

Definition 2.1.5. [116] If in an n-ary quasigroup (Q, f) code C, the operation f satisfy the check equation $f(x_1, \ldots, x_n) = e$ of this code C is an n-T-quasigroup operation, then C is called an n-T-quasigroup (Q, f) code. The element e is the additive identity of the group $(Q, +)$.

Theorem 2.1.6. [96, p. 20] *Consider $q = p^n$ where p be a prime, and let $\mathbb{E}$ be an finite extension of the field $\mathbb{F}_q$. For $\alpha \in \mathbb{E}$, the following map F is Frobenius automorphism on $\mathbb{E}$:*

$$F(\alpha) = \alpha^p$$

Theorem 2.1.7. [96, p. 20] *The distinct field automorphisms of $\mathbb{F}_{p^n}/\mathbb{F}_p$ are given by the functions $\phi_i : \mathbb{F}_{p^n} \to \mathbb{F}_{p^n}$ such that $\phi_j(\alpha) = \alpha^{p^j}$ for $\alpha \in \mathbb{F}_{p^n}$. Also, the distinct group automorphisms of $(\mathbb{F}_p, +)$ are the functions $\phi_i(x) = ix$.*

Theorem 2.1.8. [96, p. 70] *Consider n is an odd number and $a \in \mathbb{F}_q$. Then $x^n - a$ is irreducible over field $\mathbb{F}_q$ if and only if for every prime divisor r of n, a is not the r^{th} power of an element of $\mathbb{F}_q$.*

Definition 2.1.9. [96] Suppose $\mathbb{E}$ be an extension of the field $\mathbb{F}$; $Aut(\mathbb{E}/\mathbb{F})$ is the set of all automorphisms of $\mathbb{E}/\mathbb{F}$ and it forms a group with operation of function composition. It is denoted by $Gal(\mathbb{E}/\mathbb{F})$.

Remark 2.1.10. *$Gal(\mathbb{F}_{p^n}/\mathbb{F}_p) \cong \mathbb{Z}_n$.*

Theorem 2.1.11. *For any $a \in \mathbb{F}_{p^n}$, the equation $x^p - a = 0$ have a solution in $\mathbb{F}_{p^n}$.*

Proof. Consider the Frobenius automorphism $\phi : \mathbb{F}_{p^n} \to \mathbb{F}_{p^n}$ defined by $\phi(x) = x^p$. This means $\phi(\mathbb{F}_{p^n}) = \mathbb{F}_{p^n}$, i.e., $a^p = b$ for all $a \in \mathbb{F}_{p^n}$ and some $b \in \mathbb{F}_{p^n}$; conversely, for all $b \in \mathbb{F}_{p^n}$, there is some $a \in \mathbb{F}_{p^n} : a^p = b$. This means for any $a \in \mathbb{F}_{p^n}$, the equation $x^p - a = 0$ always have a solution in $\mathbb{F}_{p^n}$. □

Proposition 2.1.12. *For a prime p, $x^p - x + 1$ is irreducible over $\mathbb{F}_p$.*

Proof. Let α be a root of $f(x)$ in $\mathbb{F}_p$ and we know $\alpha^p \equiv \alpha$ in $\mathbb{F}_p$, this implies $f(x)$ has no root in $\mathbb{F}_p$. Now suppose there is some root r of f in some extension of $\mathbb{F}_p$, then it is clearly observed that all the roots of f are $\{r + k \mid k \in \mathbb{F}_p\}$. Thus if $g(x) \in \mathbb{F}_p[x]$ is an irreducible factor of $f(x)$, then $g(x) = \prod_{k \in S} (x - (r + k))$ for some subset $S \subset \mathbb{F}_p$ with cardinality $\ell < p$. On observing the coefficient of $x^{\ell-1}$ in $\mathbb{F}_p$, it implies $r \in \mathbb{F}_p$, but that is never possible. Hence, f is irreducible over $\mathbb{F}_p$. □

Theorem 2.1.13. [17, p. 148] *For a natural number n, suppose ψ denotes Euler's totient function on $\mathbb{N}$ then we have $\sum_{d|n} \psi(d) = n$.*

For more results on n-T quasigroups and automorphism of finite fields, reader may refer [96, 116].

2.2 Check equation using field $\mathbb{F}_{p^n}$

Let Q be an alphabet set of n-length vectors over $\mathbb{F}_p$. Then $Q \cong \mathbb{F}_p^n$ and the later is isomorphic to $\mathbb{F}_{p^n}$. Consider $f : Q^n \to Q$ be defined by $f(x_1, x_2, \ldots, x_n) = x_1^p + x_2^{p^2} + \cdots + x_n^{p^n}$. It can be easily verified that this map is an n-T-quasigroup operation on Q whenever p is odd. From this and the fact that $(Q, +)$ forms an Abelian group, we construct a check block equation and thereafter an n-T-quasigroup (Q, f) code in the following theorem.

Theorem 2.2.1. *Consider $Q = \mathbb{F}_{p^n}$, where p is an odd prime and the finite field automorphisms $\phi_i : Q \to Q$ such that $\phi_i(x) = x^{p^i}$ for $1 \leq i \leq n$. For an n-T-quasigroup (Q, f) code C with check equation*

$$f(x_1, x_2, \ldots, x_n) = \phi_1(x_1) + \phi_2(x_2) + \cdots + \phi_n(x_n) \tag{2.3}$$

the code $C := \{(x_1, x_2, \ldots, x_n) \in Q^n \mid f(x_1, x_2, \ldots, x_n) = 0\}$ is

- *an $\mathbb{F}_p$-subspace of Q^n, i.e., it can be viewed as a linear code over $\mathbb{F}_p$;*
- *of dimension $n(n-1)$ over $\mathbb{F}_p$;*
- *having minimum Hamming distance equal to 2.*

Proof. It can be easily observed that for $x, y \in C$ i.e., $f(x) = f(y) = 0$ implies that $f(x+y) = 0$ and $f(\alpha x) = 0$ for any $\alpha \in \mathbb{F}_p$. This implies that $x+y \in C$ and $\alpha x \in C$ by definition of the code C. This proves C is linear over $\mathbb{F}_p$.
Now, since $[\mathbb{F}_{p^n} : \mathbb{F}_p] = n$, let $\mathcal{B}_n$ denote a basis of $\mathbb{F}_{p^n}$ over $\mathbb{F}_p$. In order to construct a linearly independent set of codewords of C, fixing i^{th} position for $1 \leq i \leq n-1$, choose x_i from $\mathcal{B}_n$ other $x_i = 0$, and the last-coordinate x_n can be chosen from $\mathbb{F}_{p^n}$ to ensure (2.3) holds. Therefore, one of the basis for C is given as

$$\Big\{ (b_1, 0, \ldots, 0, -b_1^p), (b_2, 0, \ldots, 0, -b_2^p), \ldots, (b_n, 0, \ldots, 0, -b_n^p), \ldots, \\ \ldots, \left(0, \ldots, 0, b_1, -b_1^{p^{n-1}}\right), \left(0, \ldots, 0, b_2, -b_2^{p^{n-1}}\right), \ldots, \left(0, \ldots, 0, b_n, -b_n^{p^{n-1}}\right) \Big\}. \tag{2.4}$$

Thus the dimension of code C over $\mathbb{F}_p$ is $n(n-1)$. Now, in order to prove that the minimum Hamming distance of C is 2, we prove this by showing that there are no two codewords (i.e., vectors in C) at a distance of 1 from each other. For this let $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, x_2, \ldots, x_n) \in C$. Assume $x_1 \neq y_1$ so that $d_H(x, y) = 1$. Then $x, y \in C$ implies $\phi_1(x_1) + \phi_2(x_2) + \cdots + \phi_n(x_n) = 0$ and also $\phi_1(y_1) + \phi_2(x_2) + \cdots + \phi_n(x_n) = 0$. By simple subtraction of these two equations, we get $\phi_1(x_1) = \phi_1(y_1)$ which implies $x_1^p = y_1^p$ and therefore $x_1 = y_1$ since p is odd, hence $x = y$, a contradiction to our assumption. This means any two vectors with mutual Hamming distance 1 cannot be codewords of C, implying $d_H(C) \geq 2$.
Now suppose $x = (x_1, x_2, \ldots, x_n) \in C$, i.e., $f(x) = 0$, which means

$$\phi_1(x_1) + \phi_2(x_2) + \cdots + \phi_n(x_n) = 0, \text{ i.e., } x_1^p + x_2^{p^2} + \cdots + x^{p^n} = 0$$

Using Theorem 2.1.11, $x^p - x_1 = 0$ always have a solution in $\mathbb{F}_{p^n}$, suppose z_1 be the solution. Therefore we have

$$z_1^p = x_1, \text{ equivalently, } z_1^{p^2} = x_1^p \tag{2.5}$$

Consider $y = (x_2^p, z_1, x_3, x_4, \ldots, x_n)$, this vector is clearly in the code C since

$$(x_2^p)^p + z_1^{p^2} + x_3^{p^3} + \cdots + x_n^{p^n} = 0 \tag{2.6}$$

From (2.5) and (2.6), it is proven the existence of a codeword y that is at a distance of 2 from x, implying $d_H(C) = 2$. Hence, the results hold. □

Remark 2.2.2. Since the minimum Hamming distance of the defined code C in Theorem 2.2.1 is 2 implies that C can be used to detect all single errors over $\mathbb{F}_{p^n}$.

As a consequence of the above theorem, if we have a codeword $x = (x_1, x_2, \ldots, x_n) \in Q^n$, we can represent it as concatenation of n blocks each of length n over $\mathbb{F}_p$, i.e.,

$$x = (x_{11}, x_{12}, \ldots, x_{1n}, x_{21}, x_{22}, \ldots, x_{2n}, \ldots, x_{n1}, x_{n2}, \ldots, x_{nn}) \in \mathbb{F}_p^{n^2}$$

then we can detect if there is any error in the block corresponding to x_i for any i. Thus, the code defined in above Theorem can detect burst errors at n places provided all errors occur in single block.

These kind of errors are most common during any physical damage often due to a single physical disturbance or malfunction in the transmission medium. As an alternate, Reed-Solomon codes are widely being used to detect burst errors, and our designed code is much easier to implement.

Corollary 2.2.3. *In the above theorem, let π be a permutation of $\{1, 2, \ldots, n\}$ and if $\phi_i(x) = x^{p^{\pi(i)}}$ for $1 \leq i \leq n$, then it can result in the formulation of $n!$ many different such codes.*

Consider $i \in \{1, 2, \ldots, n-1\}$ and finite field $\mathbb{F}_{p^n}$ for p an odd integer. Let k be an integer such that $1 \leq k \leq n - i$. Then there are $\phi(p^i(p^k - 1))$ elements in $\mathbb{F}_{p^n}$ which have multiplicative order $p^i(p^k - 1)$. Let us define the sets $\mathcal{A}_{i,k} \subset \mathbb{F}_{p^n}$ for $i \in \{1, 2, \ldots, n\}$ as following

$$\mathcal{A}_{i,k} := \left\{ a \in \mathbb{F}_{p^n} : a^{p^i(p^k-1)} - 1 = 0 \right\}. \tag{2.7}$$

This set $\mathcal{A}_{i,k}$ comprise elements of $\mathbb{F}_{p^n}^*$ having orders p^j, $p^j(p-1)$ for $0 \leq j \leq i$ and all divisors of $p^k - 1$.

Proposition 2.2.4. *For $i \in \{1, 2, \ldots, n-1\}$ and $k \in \{1, 2, \ldots, n-i\}$, the cardinality of the set $\mathcal{A}_{i,k}$ is given by*

$$|\mathcal{A}_{i,k}| = \gcd\left(p^k - 1, p^n - 1\right).$$

Proof. This follows directly from Theorem 2.1.13. □

It is trivial to see that $|\mathcal{A}_{i,k}| \neq p^n - 1$, i.e., this set is never $\mathbb{F}_{p^n}^*$. From this, we have the following results. From now onward, in this chapter by difference in i^{th} and $(i+k)^{th}$ coordinates of $(x_1, x_2, \ldots, x_n)$, we mean $x_i - x_{i+k}$.

Now, we provide the conditions under which our error detecting code can detect double errors like transposition errors, k-jump transposition errors, twin errors,

k-jump twin errors and phonetic errors over $\mathbb{F}_{p^n}$. Each position when observed as vector over $\mathbb{F}_p$ act as block.

Theorem 2.2.5. *Suppose that the code C as defined in Theorem 2.2.1 satisfy the property that the difference of i^{th} and $(i+k)^{th}$ coordinates of all the codewords doesn't lie in the set $\mathcal{A}_{i,k}$. Then the following holds:*

(*a*) *The code C can be used to detect the k-jump transposition error at places $(i, i+k)$.*

(*b*) *The code C always detect the twin error at places $(i, i+k)$.*

Proof. For part (a), consider $x = (x_1, \ldots, x_i, \ldots, x_{i+k}, \ldots, x_n) \in C$ and $x_i \neq x_{i+k}$, for some $i \in \{1, \ldots, n-k\}$, then we have

$$\phi_1(x_1) + \cdots + \phi_i(x_i) + \cdots + \phi_{i+k}(x_{i+k}) + \cdots + \phi_n(x_n) = 0 \tag{2.8}$$

and, let $y = (x_1, \ldots, x_{i+k}, \ldots, x_i, \ldots, x_n) \in C$, i.e., the k-jump transposition at $(i, i+k)$ place of codeword x. Then we have

$$\phi_1(x_1) + \cdots + \phi_i(x_{i+k}) + \cdots + \phi_{i+k}(x_i) + \cdots + \phi_n(x_n) = 0 \tag{2.9}$$

By solving (2.8) and (2.9), we get $\phi_i(x_i - x_{i+k}) + \phi_{i+k}(x_{i+k} - x_i) = 0$ which implies $(x_i - x_{i+k})^{p^i} + (x_{i+k} - x_i)^{p^{i+k}} = 0$, and reduces to $(x_i - x_{i+k})^{p^i}[1 - (x_i - x_{i+k})^{p^i(p^k-1)}] = 0$. Since $x_i \neq x_{i+k}$ is assumed, $(x_i - x_{i+k})^{p^i(p^k-1)} = 1$. Supposing $x_i - x_{i+k} = z \neq 0$, it implies that there exists an element $z \in \mathbb{F}_{p^n}^*$ such that $z^{p^i(p^k-1)} = 1$. That means if there exist element $z \in \mathcal{A}_{i,k}$, difference in the i^{th} and $(i+k)^{th}$ coordinates lie in set $\mathcal{A}_{i,k}$ then code C is not able to detect k-jump transposition error at places $(i, i+k)$.

For part (b), consider $x = (x_1, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_{i+k-1}, a, x_{i+k+1}, \ldots, x_n) \in C$ and $x_i \neq a$ for some i, then we have

$$\phi_1(x_1) + \cdots + \phi_i(a) + \cdots + \phi_{i+k}(a) + \ldots + \phi_n(x_n) = 0 \tag{2.10}$$

and, $y = (x_1, \ldots, x_{i-1}, b, x_{i+1}, \ldots, x_{i+k-1}, b, x_{i+k+1} \ldots, x_n) \in C$ and $a \neq b$, i.e., the twin error at $(i, i+k)$ place of codeword x, we have

$$\phi_1(x_1) + \cdots + \phi_i(b) + \cdots + \phi_{i+k}(b) + \cdots + \phi_n(x_n) = 0 \tag{2.11}$$

By solving (2.10) and (2.11), we get $\phi_i(a-b) + \phi_{i+k}(a-b) = 0$ which implies $(a-b)^{p^i} + (a-b)^{p^{i+k}} = 0$ and reduces to $(a-b)^{p^i}[1 - (b-a)^{p^i(p^k-1)}] = 0$

As $a \neq b, (b-a)^{p^i(p^k-1)} = 1$ implies that there exist $b - a = z \neq 0$ and $z \in \mathbb{F}_{p^n}^*$ such that $z^{p^i(p^k-1)} = 1$. That means if there exist element $z \in \mathcal{A}_{i,k}$, difference in the i^{th} and $(i+k)^{th}$ coordinates lie in set $\mathcal{A}_{i,k}$ then code C not able to detect twin error at places $(i, i+k)$. □

The following results hold true for $k = 1$, and for detecting the transposition error and twin errors at adjacent places (as coordinates in $\mathbb{F}_{p^n}$).

Theorem 2.2.6. *Suppose that the code C as defined in Theorem 2.2.1 satisfy the property that the difference of i^{th} and $(i+1)^{th}$ coordinates of all the codewords doesn't lie in the set $\mathcal{A}_{i,1}$. Then the following holds:*

(*a*) *The code C can be used to detect an adjacent transposition error at places $(i, i+1)$.*

(*b*) *The code C always detect the twin error at places $(i, i+1)$.*

Proof. The proof is similar as above Theorem 2.2.5, for particular value $k = 1$. □

In order to detect phonetic errors, we first establish the cases where the polynomial $x^p - x + 1$ becomes irreducible over finite extensions of $\mathbb{F}_p$.

Theorem 2.2.7. *For $p \leq n$ being a prime number, $f(x) = x^p - x + 1 \in \mathbb{F}_{p^n}[x]$ is irreducible if and only if $p \nmid n$.*

Proof. For the forward part, suppose $p \nmid n$, then $\mathbb{F}_{p^p} \sqsubseteq \mathbb{F}_{p^n}$ Since $f(x)$ is irreducible over $\mathbb{F}_p$ using Proposition 2.1.12, $f(x)$ splits in $\mathbb{F}_{p^p}$, which implies $f(x)$ is reducible over $\mathbb{F}_{p^n}$, a contradiction.
Conversely, suppose $f(x)$ is reducible over $\mathbb{F}_{p^n}$, then $f(x) = f_1(x)f_2(x)\dots f_k(x)$ be an irreducible factorization over $\mathbb{F}_{p^n}$, where $\deg f_i = d_i < p$ for each i. Then $f_i(x)$ splits in $\mathbb{F}_{p^{nd_i}}$ for each i, and therefore $f(x)$ splits in $\mathbb{F}_{p^{nd}}$ where $d = \text{lcm}\{d_i : 1 \leq i \leq k\}$. Again, since $f(x)$ is irreducible over $\mathbb{F}_p$, it splits in $\mathbb{F}_{p^p}$. Thus, $\mathbb{F}_{p^p} \sqsubseteq \mathbb{F}_{p^{nd}}$, which implies $p \mid nd$. This means $p \mid d$, which is impossible since $d_i < p$ for each i, if $p \mid d$ then $p \mid d_i$ for some i, a contradiction. Hence, the result holds. □

Theorem 2.2.8. *Suppose $p \nmid n$, then the code C can be used to detect the phonetic errors in codewords at places $(i, i+1)$.*

Proof. Consider $x = (x_1, \dots, x_{i-1}, c, 0, x_{i+2}, \dots, x_n) \in C, c \neq \{0, 1\} \subset \mathbb{F}_{p^n}$. Thus,

$$\phi_1(x_1) + \phi_2(x_2) + \cdots + \phi_{i-1}(x_{i-1}) + \phi_i(c) + \phi_{i+1}(0) + \phi_{i+1}(x_{i+1}) + \cdots + \phi_n x_n = 0 \tag{2.12}$$

and if $y = (x_1, x_2, \ldots, x_{i-1}, 1, c, x_{i+2}, \ldots, x_n) \in C$, i.e., phonetic error at $(i, i+1)$ places of codeword x, then we have

$$\phi_1(x_1) + \phi_2(x_2) + \cdots + \phi_{i-1}(x_{i-1}) + \phi_i(1) + \phi_{i+1}(c) + \phi_{i+1}(x_{i+1}) + \cdots + \phi_n(x_n) = 0 \tag{2.13}$$

By solving (2.12) and (2.13), we get $\phi_i(c) + \phi_{i+1}(0) = \phi_i(1) + \phi_{i+1}(c)$ implies $c^{p^i} = 1 + c^{p^{i+1}}$ and then we get $c^{p^{i+1}} - c^{p^i} + 1 = 0$. Put $c^{p^i} = t$ then, $t^p - t + 1 = 0$. By Theorem 2.2.7, $x^p - x + 1$ is irreducible over $\mathbb{F}_{p^n}$ if $p \nmid n$. So, code C detects all the phonetic errors at places $(i, i+1)$ if $p \nmid n$. □

The present study includes a toy example to showcase the error detection potential of the code C as outlined in Theorem 2.2.1. In particular, we consider a transmission scenario where the 2^{nd} place of codewords is frequently interchanged with the 10^{th}, 9^{th}, or 7^{th} positions, while the 4^{th} position is commonly replaced with the 6^{th}. It is shown that the proposed code is capable of detecting such errors with minimal expense.

Example 2.2.9. Consider $\mathbb{F}_{3^{10}} = \mathbb{F}_3[x]/\langle x^{10} + 2x^6 + 2x^5 + 2x^4 + x + 2\rangle$. Then from (2.7), it follows that

- For $i = 2, k = 8$, the set $\mathcal{A}_{i,k}$, having 8 elements, is given by
$$\begin{cases} 2x^9 + 2x^7 + x^6 + 2x^4 + 2x^3 + x^2 + 2x \\ 2x^9 + 2x^7 + x^6 + 2x^4 + 2x^3 + x^2 + 2x + 1 \\ x^9 + x^7 + 2x^6 + x^4 + x^3 + 2x^2 + x + 1 \\ 2 \\ x^9 + x^7 + 2x^6 + x^4 + x^3 + 2x^2 + x \\ x^9 + x^7 + 2x^6 + x^4 + x^3 + 2x^2 + x + 2 \\ 2x^9 + 2x^7 + x^6 + 2x^4 + 2x^3 + x^2 + 2x + 2 \\ 1 \end{cases}$$
- For $i = 2, k = 5$, the set $\mathcal{A}_{i,k}$ have 242 elements.
- For $i = 2, k = 7$, the set $\mathcal{A}_{i,k}$ have 2 elements $\{1, 2\}$.
- For $i = 4, k = 2$, the set $\mathcal{A}_{i,k}$ have 2 elements $\{1, 2\}$.

Consider the $\mathbb{F}_3$-linear code C having length 10 as defined in Theorem 2.2.1, suppose we have a codeword

$$(1, x, -1, x^2, x^3, 1-x, x, -x^2, x^3, 2x^9+x^8+x^7+x^6+x^5+x^4+2x^3+x^2+2x+1) \in C$$

Here, the difference of 2^{nd} and 8^{th} is $x + x^2 \notin \mathcal{A}_{2,8}$. Thus flip the second and eighth coordinate, the resultant vector will not be a legitimate codeword, since following

(2.3) it is easy to observe that $f(x_1, x_2, \ldots, x_{10}) = 2x^9 + x^7 + x^4 + x^3 + 2x + 2 \neq 0$. Similar results hold for other cases as well.

Consider an another $\mathbb{F}_3$- linear code C having length 10 as defined in Theorem 2.2.1, suppose we have a codeword

$$(1, 1 + x^2, x^2, x^3, -x^2, x^3, 1 + x + x^2, 1 + x^3, x, -x^3 - 1) \in C$$

and, suppose it has the twin errors at coordinates $(4, 6)$, the received word is

$$(1, 1 + x^2, x^2, x, -x^2, x, 1 + x + x^2, 1 + x^3, x, -x^3 - 1) \in C$$

Here, the difference in the coordinates $(4, 6)$ of the actual and received codeword is $x - x^3 \notin \mathcal{A}_{4,6}$. Thus, if the twin error occur at the fourth and seventh coordinates, the resultant vector will not be legitimate codeword.

2.3 Check equation using group $\mathbb{F}_p$

Suppose $\phi_1, \phi_2, \ldots, \phi_n$ be automorphisms of $(\mathbb{F}_p, +)$ then the n-ary linear quasigroup operation on $\mathbb{F}_p$ as $f(x_1, x_2, \ldots, x_n) = \phi_1(x_1) + \phi_2(x_2) + \cdots + \phi_n(x_n)$ when serves as check equation, produces an n-T-quasigroup code.

Proposition 2.3.1. *Consider $Q = \mathbb{F}_p$, p an odd prime, $n = p - 1$ and the n automorphisms $\phi_i : Q \to Q$ such that $\phi_i(x) = ix$ for $1 \leq i \leq n - 1$. An n-T-quasigroup (Q, f) code with check equation*

$$f(x_1, x_2, \ldots, x_n) = \phi_1(x_1) + \phi_2(x_2) + \cdots + \phi_n(x_n), \tag{2.14}$$

the code $C := \{(x_1, x_2, \ldots, x_n) \in Q^n \mid f(x_1, x_2, \ldots, x_n) = 0\}$ is

- *an $\mathbb{F}_p$-subspace of Q^n, i.e., a linear code over $\mathbb{F}_p$;*
- *of dimension $n - 1$ over $\mathbb{F}_p$;*
- *having minimum Hamming distance equal to 2.*

Proof. It is easy to observe that C is linear over $\mathbb{F}_p$. In order to construct a linearly independent set of codewords of C, consider the set

$$\{(1, 0, 0, \ldots, 0, 1), (0, 1, 0, \ldots, 0, 1 - n), (0, 0, 1, \ldots, 0, 1 - 2n), \\ \ldots, (0, 0, \ldots, 1, 1 - (n - 2)n)\}.$$

It is easy to observe that this set spans C. Hence, $\dim C = n - 1$. Now, in order to prove that the minimum Hamming distance of C is 2, we prove this by showing

that there are no two codewords (i.e., vectors in C) at a distance of 1 from each other. For this let $x = (x_1, x_2, \cdots, x_n)$ and $y = (y_1, x_2, \ldots, x_n) \in C$. Assume $x_1 \neq y_1$ so that $d_H(x, y) = 1$. Then $x, y \in C$ implies $\phi_1(x_1) + \phi_2(x_2) + \cdots + \phi_n(x_n) = 0$ and also $\phi_1(y_1) + \phi_2(x_2) + \cdots + \phi_n(x_n) = 0$. By simple subtraction of these two equations, we get $\phi_1(x_1) = \phi_1(y_1)$ which implies $1 \cdot x_1 = 1 \cdot y_1$ and therefore $x_1 = y_1$ since p is odd, hence $x = y$, a contradiction to our assumption. This means any two vectors with mutual Hamming distance 1 cannot be codewords of C, implying $d_H(C) \geq 2$. Now suppose $x = (x_1, x_2, \ldots, x_n) \in C$, i.e., $f(x) = 0$, which means

$$\phi_1(x_1) + \phi_2(x_2) + \cdots + \phi_n(x_n) = 0, \ i.e., 1x_1 + 2x_2 + \cdots + nx_n = 0$$

As, we know $2x = x_1$ always has solution in finite field $\mathbb{F}_p$, $p \geq 3$, suppose z_1 be the solution. Therefore we have $2z_1 = x_1$. Now, consider $y = (2x_2, z_1, x_3, x_4, \cdots, x_n)$, this is clearly a valid codeword of C since

$$2x_2 + 2z_1 + 3x_3 + 4x_4 + \cdots + nx_n = 0. \tag{2.15}$$

Therefore, the existence of a codeword y that is at a distance of 2 from x is ensured, implying $d_H(C) = 2$. Hence, the results holds. □

Remark 2.3.2. Since the minimum Hamming distance of the defined C is 2, it can be used to detect all single errors.

Theorem 2.3.3. *Suppose code C be as defined in Proposition 2.3.1. Then the code C can be used to detect:*

- *all transposition errors at places $(i, i+1)$;*
- *all k-jump transposition errors at places $(i, i+k)$ for $i \in \{1, \ldots, n\}$ and $1 \leq k \leq n - i$;*
- *all twin errors at places $(i, i+k)$ except at the coordinates 'i' where $2i \equiv (p-k) \mod p$ for $i \in \{1, \ldots, n\}$ and $1 \leq k \leq n - i$;*
- *all twin errors at places $(i, i+1)$ except at the coordinates 'i' where $2i \equiv (p-1) \mod p$;*
- *all the phonetic errors $c0 \to 1c$, $c \neq 0, 1$ except if $c \equiv (p-i) \mod p$.*

Proof. In order to detect k-jump transposition error, consider $x = (x_1, x_2, \ldots, x_i, \ldots, x_{i+k}, \ldots, x_n) \in C$ and $x_i \neq x_{i+k}$, for some $i \in \{1, \ldots, n\}$, then we have

$$\phi_1(x_1) + \phi_2(x_2) + \cdots + \phi_i(x_i) + \cdots + \phi_{i+k}(x_{i+k}) + \cdots + \phi_n(x_n) = 0 \tag{2.16}$$

and, let $y = (x_1, x_2, \ldots, x_{i+k}, \ldots, x_i, \ldots, x_n) \in C$, i.e., transposition at $(i, i+k)$ place of codeword x. Then we have

$$\phi_1(x_1) + \phi_2(x_2) + \cdots + \phi_i(x_{i+k}) + \cdots + \phi_{i+k}(x_i) + \cdots + \phi_n(x_n) = 0. \tag{2.17}$$

By solving (2.16) and (2.17), we get $\phi_i(x_i) - \phi_i(x_{i+k}) + \phi_{i+k}(x_{i+k}) - \phi_{i+k}(x_i) = 0$ which implies $ix_i - ix_{i+k} + (i+k)x_{i+k} - (i+k)x_i = 0$. Thus on evaluation, we get $x_i = x_{i+1}$ which is a contradiction. So, code C detects all the transposition error at places $(i, i+k)$.
For the detection of twin errors, consider $x = (x_1, \ldots, a, \ldots, a, , \ldots, x_n) \in C$ and $x_i \neq a$ for some i, then we have

$$\phi_1(x_1) + \cdots + \phi_i(a) + \cdots + \phi_{i+k}(a) + \cdots + \phi_n(x_n) = 0 \tag{2.18}$$

and, $y = (x_1, \ldots, b, \ldots, b, \ldots, x_n) \in C$ and $a \neq b$, i.e., the twin error at $(i, i+k)$ place of codeword x, we have

$$\phi_1(x_1) + \cdots + \phi_i(b) + \cdots + \phi_{i+k}(b) + \cdots + \phi_n(x_n) = 0. \tag{2.19}$$

By solving (2.18) and (2.19), we get $\phi_i(a) - \phi_i(b) + \phi_{i+k}(a) - \phi_{i+k}(b) = 0$ which implies $ia - ib + (i+k)a - (i+k)b = 0$, $(a-b)(2i+k) = 0$ implies $a = b$ if $2i \neq -k \bmod p$ implies $2i \not\equiv (p-k) \bmod p$. So, code C can detect all the twin error at places $(i, i+k)$ except at places when $2i \equiv (p-k) \bmod p$. For $k = 1$, all the twin errors will be detected at coordinates $(i, i+1)$ except where $2i \equiv (p-1) \bmod p$.
Finally, in order to detect phonetic errors, consider $x = (x_1, \ldots, x_{i-1}, c, 0, x_{i+2}, \ldots, x_n) \in C$ and $c \neq 0, 1$, then we have

$$\phi_1(x_1) + \cdots + \phi_{i-1} + \phi_i(c) + \phi_{i+1}(0) + \phi_{i+1} + \cdots + \phi_n(x_n) = 0 \tag{2.20}$$

and, $y = (x_1, \ldots, x_{i-1}, 1, c, x_{i+2}, \ldots, x_n) \in C$, i.e., the phonetic error at $(i, i+1)$ place of codeword x, we have

$$\phi_1(x_1) + \cdots + \phi_{i-1}(x_{i-1}) + \phi_i(1) + \phi_{i+1}(c) + \phi_{i+2}(x_{i+2}) + \cdots + \phi_n(x_n) = 0. \tag{2.21}$$

By solving (2.20) and (2.21), we get $\phi_i(c) - \phi_i(1) + \phi_{i+1}(0) - \phi_{i+1}(c) = 0$ which implies $c = -i \equiv (p-i) \bmod p$. So, code C is not able to detect all the phonetic errors at places $(i, i+1)$ if $c \equiv (p-i) \bmod p$. Hence, the result holds. □

2.4 Comparative analysis and applications

Proposition 2.4.1. *The maximum number of $\mathbb{F}_{p^n}$-operations for detecting if there is any single 'position' error in any valid codeword of C, as defined in Theorem 2.2.1, is given by $\mathcal{O}(n \log_2 p)$.*

Proof. In order to compute value of $\phi_i(a)$ for any $a \in \mathbb{F}_{p^n}$, the multiplication operations performed over $\mathbb{F}_{p^n}$ is at most $\sum_{i=1}^{n} (i \log_2 p)$. Now for validation of check equation (2.3), we need to do n additions. Hence, the result follows immediately. □

On the other hand, suppose we consider a Reed-Solomon code $[n, k]$, with n be the size of codeword and k be the number of information symbols in the codeword. To detect an error in a codeword using Peterson-Gorenstein-Zierler decoding, we need to calculate the syndrome polynomial which involves evaluating the received polynomial at $n - k$ distinct field elements. This requires $2(n-1)(n-k)$ $\mathbb{F}_{p^n}$-operations [110]. This value is quadratic in n, whereas, the code that we proposed in Theorem 2.2.1 has lower complexity, thus serves as a better candidate for error detection purposes.

2.4.1 ISBN code

Now we present a generalization of the check digit systems proposed by various authors in [25, 26, 97] for the ISBN code. Specifically, we extend the applicability of these systems to any odd prime, rather than just the prime $p = 11$ or 13 used in previous works. The proposed check digit system in Proposition 2.3.1 is capable of detecting both single and double errors of types $(1) - (5)$ as defined in Table 2.1, and encompasses all the previously reported results mentioned in [97]. Thus, we provide a more comprehensive and versatile solution for error detection than ISBN codes.

Consider a finite field $\mathbb{F}_p$ and the automorphisms $\phi_i : \mathbb{F}_p \to \mathbb{F}_p$ as $\phi_i(x) = ix$ for $i \in \{1, \ldots, p-1\}$. Then, (2.14) implies $f(x_1, x_2, \ldots, x_{p-1}) = \sum_{i=1}^{p-1} ix_i \pmod p$. Using this f, we can define $(p-1)$-T-quasigroup code. This system detects all the single position errors, k-jump transposition errors and k-jump twin errors except at the coordinates $(i, i+k)$ where 'i' follows $2i \equiv p - k \pmod p$ and phonetic error $c0 \to 1c,\ c \neq 0, 1$ except when $c \equiv (p - i) \pmod p$.

As a consequence of this, if we take $p = 11$, then we have 10-digit ISBN code

over $\mathbb{F}_{11}$ with check equation

$$\sum_{i=1}^{10} i \cdot x_i \equiv 0 \pmod{11}. \tag{2.22}$$

The system (2.22) detects all type of errors of types (1)–(5) except the twin error at places $(5, 6)$, since $2i \equiv 10 \pmod{11}$ when $i = 5$.

Also, the proposed system is able to detect all the phonetic errors except when $c \equiv (11 - i) \pmod{11}$ i.e., it cannot detect the phonetic errors like:

$10, 0 \to 1, 10$ on the place (1,2)	$9, 0 \to 1, 9$ on the place (2,3)
$8, 0 \to 1, 8$ on the place (3,4)	$7, 0 \to 1, 7$ on the place (4,5)
$6, 0 \to 1, 6$ on the place (5,6)	$5, 0 \to 1, 5$ on the place (6,7)
$4, 0 \to 1, 4$ on the place (7,8)	$3, 0 \to 1, 3$ on the place (8,9)
$2, 0 \to 1, 2$ on the place (9,10)	

A modification made to the ISBN-10 code [97] over $\mathbb{F}_{11}$, this improves the ISBN-10 and detects all errors of types (1)–(5) without exception.

$$\sum_{i=1}^{5} i \cdot x_i + \sum_{j=6}^{10} (5 - j) \cdot x_j \equiv 0 \pmod{11} \tag{2.23}$$

Similarly, for the proposed system we modify the system over $\mathbb{F}_p$, p-odd prime as described in (2.24):

$$\sum_{i=1}^{(p-1)/2} i \cdot x_i + \sum_{j=\frac{p+1}{2}}^{p} \left(\frac{p-1}{2} - j \right) \cdot x_j \equiv 0 \pmod{p} \tag{2.24}$$

This modification can be employed for any prime p to the code provided in Proposition 2.3.1, and it effectively eliminates all exceptions in error detection. Undoubtedly, this research represents a significant advancement and holds great potential for making a profound impact.

2.4.2 Check digit system to detect an ineligible cheque

Now we discuss the check digit system for bank routing numbers, which is a method for error detection and preventing fraud in financial transactions. Figure 2.1, shows a typical 9-digit bank routing number. The first four digits represent the federal reserve routing symbol, the next four digits identify the bank or financial institution where the account is held, and the last digit is check digit. The check

digit is calculated using a specific algorithm, like the Verhoeff algorithm [124], that takes into account the other digits in the routing number. This check digit helps detect errors (e.g. typos during data entry) that might occur in the routing number. Similar to check digits, we can design a check block system using the proposed scheme. Consider the case where $p = 3$ and $n = 3$, we can design a linear code with a check block that that is a linear combination of the other blocks, allowing for error detection in the original blocks. For instance, we could use a linear code C as described in Theorem 2.2.1 over $\mathbb{F}_{3^3}$ as follows:

$$C = \{(x_1, x_2, x_3) \in \mathbb{F}_{3^3}^3 \mid f(x_1, x_2, x_3) = \phi_1(x_1) + \phi_2(x_2) + \phi_3(x_3) = 0\} \quad (2.25)$$

where $x_1 = (x_{11}, x_{12}, x_{13}), x_2 = (x_{21}, x_{22}, x_{23})$ and $x_3 = (x_{31}, x_{32}, x_{33})$. By utilizing the code specified in (2.25), if any error occurs during transmission or data entry, the check block will not match the sum of the other blocks, indicating that an error has occurred. This can also be used for various applications where error detection and prevention are essential, such as in computer networks and financial transactions.

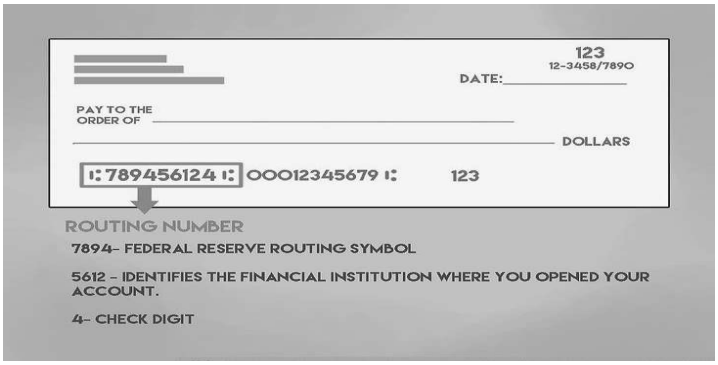

Figure 2.1: Cheque of federal reserve bank

Thus, the proposed check digit/ block system is easy to implement rather then previous check digit systems for detecting the burst errors in the routing number for an ineligible cheque.

2.4.3 Social security number

In the United States, a unique nine-digit number known as the *Social Security number* (SSN) has been assigned to every citizen. These numbers are utilized for a variety of purposes, including taxation and personal identification. Similarly, in countries such as India, an *Aadhaar number* is used to establish a person's identity.

We can design a check block system similar to SSN using a code defined over

finite field $\mathbb{F}_{p^n}$. For instance, lets consider the case where $p = 3$ and $n = 3$, and let C be the linear code as described in Theorem 2.2.1 over $\mathbb{F}_{3^3}$:

$$C = \{(x_1, x_2, x_3) \in \mathbb{F}_{3^3}^3 \mid f(x_1, x_2, x_3) = \phi_1(x_1) + \phi_2(x_2) + \phi_3(x_3) = 0\} \quad (2.26)$$

where $x_1 = (x_{11}, x_{12}, x_{13}), x_2 = (x_{21}, x_{22}, x_{23})$ and $x_3 = (x_{31}, x_{32}, x_{33})$. The purpose of this code is to detect errors and prevent frauds, similar to how SSN is used for identification and security purposes.

This idea can be generalized over any finite field $\mathbb{F}_{p^n}$, by defining a linear code that satisfies certain properties to ensure error detection and prevention. For example, one could define a code that includes a check block consisting of a linear combination of the other blocks, which can be used to detect errors in the original blocks. These types of codes are commonly used in computer science and engineering for error detection and correction in data transmission and storage.