Mathematics Interdisciplinary Research **7** (2022) 301 – 329

*Review Article*

# The Applications of Algebraic Polynomial Rings in Satellite Coding and Cryptography

*Amir Bagheri* [*] *and Hassan Emami*

**Abstract**

This survey illustrates and investigates the application of polynomial rings over finite fields to generate PRN codes for Global Navigation Satellite System (GNSS) satellites. In GNSS, satellites continually broadcast signals at two or more frequencies, including pseudo-random noise (PRN) codes. Each GNSS satellite has its own PRN code, and due to the unique mathematical properties of PRN codes, all satellites can communicate at the same frequency without interfering with another one. Although the PRN code appears to be devoid of any discernible structure, it is composed of a deterministic series of pulses that will repeat itself after its period. The PRN code generator employs two shift registers known as Gold polynomials, and the suitable polynomial is decided by the number of satellites. The approach used in satellites is based on the usage of two primitive polynomials, with the output of the first polynomial being used as input for the second polynomial.

Keywords: polynomial rings, finite fields, PRN codes.

2020 Mathematics Subject Classification: 11T06.

**How to cite this article**
A. Bagheri and H. Emami, Applications of algebraic polynomial rings in satellite coding and cryptography, *Math. Interdisc. Res.* **7** (2022) 301 – 329.

## 1. Introduction

Mathematics is utilized in a thorough study of many disciplines and other sciences as a branch of the basic sciences. Mathematics does not rely heavily on other disciplines; that is, while certain natural sciences and engineering utilize mathematics,

[*] Corresponding author (E-mail: a_bageri@tabrizu.ac.ir)

Academic Editor: Seyed Hassan Alavi
Received 12 April 2022, Accepted 3 August 2022
DOI: 10.22052/mir.2022.246273.1345

mathematics uses just its fundamental axioms and facts to establish a mathematical statement and does not require other sciences. Discussing the utility of mathematics in other sciences in the past, clear emphasis was placed on applied mathematics. Nowadays and with dramatic development, pure mathematics, for example, algebra has many useful applications in other sciences.

The application of mathematics in other disciplines, of course, increases the significance of this subject. Today, the applications of mathematics in other disciplines are not only limited to physics and chemistry, but it is also used in other fields of technical and engineering, and even the role of mathematics in the growth of social and human sciences is essential [29]. It is worth noting that for many years it was assumed that only the practical and computational aspects of mathematics could be applied in other sciences, but today many of the pure and abstract branches of mathematics, such as algebra, analysis, etc. have very important applications in other sciences.

The concept of abstract polynomial rings is a single mathematical concept that is extensively used in a variety of disciplines nowadays. Groups, rings, and fields are algebraic structures formed by integrating a set and one (or two) operation(s). The study of these structures, and notably their application in other fields, especially in coding theory, has made significant progress, to the point where specific polynomials are employed as code generators. Two of the most important mathematical algebraic ideas in its development are the cipher and encryption theory.

Furthermore, there are Galois groups and mathematical ideas that are utilized in the construction of code and encryption. Coding theory, or especially algebraic coding theory, is the study of techniques for transferring information from one location to another efficiently and accurately manner. The theory has been developed for a wide range of applications, including the reduction of noise in compact disc recordings, the transmission of financial information over telephone lines, data transfer from one computer to another or from memory to the central processor, and information transmission from a remote source, such as a satellite or spacecraft that sends images or signals to Earth [23].

The followings are instances of polynomial study in various applications of coding and encryption theory. Massey, [17], employs both traditional and current algebraic techniques such as finite fields, group theory, and polynomial algebra. The findings of coding theory allow for the development of trustworthy techniques for storing and/or transferring data from malfunctioning systems. A channel is a physical medium via which information is conveyed. Channels include telephone wires and the environment, such as the atmosphere. Unwanted interruptions, known as noise, might cause the information received to differ from what was delivered ([3]). In fact, the only control we have over this noise is the selection of a reliable transmission channel and the use of various noise filters to overcome various forms of interference that may be encountered. These are engineering issues.

Once we've decided on the optimum mechanical solution for addressing these issues, we may move on to the design of the encoder and decoder. Puchinger and

Wachter-Zeh in [21] examine fast algorithms for linearized polynomial operations and offered a novel multiplication method for skew polynomials with sub-quadratic complexity in polynomial degrees, irrespective of the underlying field extension degree $m$.

Their findings demonstrate how the novel rapid operations on linearized polynomials lead to the first error and erasure decoding method for Gabidulin codes of sub-quadratic complexity. Ivchenko et al. [13] investigate the analytic, number-theoretic, and asymptotic characteristics of Krawtchouk polynomials under various parameter relationships, as well as the probabilistic features of polynomials with random parameters. The Krawtchouk polynomials are used to study the cryptographic characteristics of Boolean functions and coding theory. Permutation polynomials and their inverses are studied by Zheng et al. [30] because they have applications in cryptography, coding theory, and combinatorial design theory. Flaut [4] demonstrates several applications of a difference equation of degree $k$. Based on such an equation, he provides a mechanism for message encryption and decryption.

Furthermore, they show how to use the matrix associated with a difference equation of degree $k$ in coding theory. In a study, Mesnager [18] chose two major classes of Boolean functions with ancurved and semi-curved functions based on their algebraic and compositional properties. They demonstrate that oval polynomials (which are connected to projective plane hyperovals) give birth to numerous new constructions of infinite classes of semi-bent Boolean functions in even dimension.

Paryasto et al., [20], offer an implementation of an efficient technique to convert from the representation of a field element in one basis to the representation of a field element in another basis in polynomials. It is feasible to use this approach to expand an implementation on one basis such that it supports additional basis choices. Arnault et al., [1], began by recalling the conventional matrix form of linear finite state machines (LFSMs). They then offered a new matrix representation with polynomial fractional coefficients. In the second section, a novel design criteria for LFSRs termed diffusion delay is proposed and thoroughly contrasted with existing related ideas. Thus, they offered a novel technique to randomly choose LFSRs with desirable features (including the new one) and sparse descriptions dedicated to hardware and software designs utilizing the matrix form.

Hell [12] examined a family of weak feedback polynomials for LFSRs in the nonlinear combiner. This class of weak polynomials was established in 2004, and the attack's primary feature is that the noise variables are represented as vectors. Using coding theory, he assesses the attack's complexity. He demonstrates that polynomial groups may be seen as generating polynomials for a convolutional code. The challenge of determining the attack complexity is thus identical to determining the minimal row distance of the associated generating matrix. Guo and Fu, [7], offered a new approach for disguising the algebraic structure of linear codes used in code-based encryption in research. They presented the so-called semi-linear transformations in coding theory, do extensive research on their algebraic characteristics, and then creatively apply them to the design of code-based cryptosystems. Their solution, as compared to some existing code-based cryptosystems, allows for significantly more compact encoding of public keys. In [30], Zheng demonstrates how permutation polynomials may be used in cryptography, coding theory, combinatorial designs, and other fields of mathematics and engineering.

Two new classes of permutation polynomials over finite fields are introduced as a result of their finding. Other novel permutation polynomials are developed based on the connection between equivalent equations and permutation polynomials. In addition, Gulak in [6] offered a method for creating primitive polynomials, which are utilized in the construction of radio engineering systems, critical infrastructure control systems, and other socially relevant information systems.

Such polynomials, in particular, can be utilized to build cryptographic scheme elements such as pseudo-random number generators, guaranteed period nodes, and replacement nodes (substitution).

Muchtadi-Alamsyah, [19], gave a study of algebraic structures in cryptography, such as groups, rings, and fields. Some of his results include the conversion of polynomial bases to normal bases and the identification of a weak class of elliptic curves that are unsuitable for cryptography. They will also briefly discuss how higher algebraic structures are used in cryptography and coding theory.

We have already briefly given a review of prior research about the applications of algebraic polynomials in various codings and generally coding theory. In this article, we are going to take a fresh look at some of the most significant uses of the polynomial ring and primitive polynomials. Indeed, in this work we investigate the application of the polynomial ring on finite fields in the generation of various satellite codes and their transmission to Earth to establish coordinates and other purposes. We examine the subject in its simple instance and for finite sets, with the goal of introducing the applications of these algebraic ideas in satellite-to-terrestrial communication.

The structure of this document is as follows: First, key algebraic concepts and examples will be introduced. The goal of this article is to introduce a very strong application of polynomial rings in coding theory in an interdisciplinary manner. So we introduce the basic mathematical preliminaries in a very simple way for those who are familiar with geomatics and coding theory. Also, for the people who are familiar with mathematics, we provide some questions to research. This is the reason that we put the simple definitions and theorems (they can skip the next section). For simplicity, we ignore all of the proofs and just introduce the necessary definitions in the next section. Then, we will quickly present a variety of codes, notably those generated by polynomials. Afterward, satellite systems will be introduced, which have the task of communicating with ground equipment to identify the position and coordinates of the points.

## 2. Algebraic Prerequisites

In mathematics, several algebraic structures are utilized. Particularly in the subject of set theory, the most significant things are called groups, rings, and fields. We begin with some essential definitions and assumptions that will be used throughout the paper.

**Definition 2.1.** Let $G$ be a non-empty set equipped with a binary operation. The algebraic structure $(G, *)$ is called a group whenever: $G$ is closed under $*$, $*$ is associative on $G$, there is an identity element in $G$ with respect to $*$ and every element of $G$ has inverse. Also $G$ is said to be an abelian group if $*$ is commutative.

If $G$ is an infinite set that makes a group with an operation, then we say that $G$ is an infinite group; Otherwise $G$ is called a finite group. In the case that $G$ is finite, the number of its elements is called the order of $G$. The groups that will be used in this article will be finite groups. For example, if $G$ is the set of the integer remainders modulo positive integer $n$, then this set makes a group with a mathematical operation.

Obviously, if we divide any integer by a positive integer $n$, then the remainder is like $r$ such that it satisfies the inequality $0 \leq r < n$. Indeed, these residues which individually represent all the integers that are the remainder of their division by $n$, form a group denoted by $\mathbb{Z}_n$. It is clear that if $n = 2$, then the group is equal to $\{0, 1\}$, which is highly useful and well-known.

**Definition 2.2.** Let $G$ be a group, and let $H$ be a subset of $G$. If $H$ is also a group by the operation of $G$, we say that $H$ is a subgroup of $G$. $G$ is said to be a cyclic group if it consists of powers of some elements of $G$. In other words, $G$ is a cyclic group if and only if there is an element $a$ of $G$ such that:

$$G = \{e, a^{\pm 1}, a^{\pm 2}, \ldots\}.$$

$a$ is called a generator of this cyclic group. If a positive power of $a$ equals the identity element, then a finite cyclic group is obtained:

$$G = \{e, a, a^2, \ldots, a^{n-1}\}.$$

Rings are another algebraic structures that differ from groups. The following definition introduces the concept of rings.

**Definition 2.3.** Assume that $R$ is a non-empty set with two binary operations $*$ and $\Delta$ on it. The algebraic structure $(R, *, \Delta)$ is called a ring whenever: $(R, *)$ is an abelian group, it is closed and associative with respect to $\Delta$ and $\Delta$ distributes over the first operator $*$.

Conventionally, the first operator is called addition, and the second operator is said to be multiplication. In general, we show them by $+$ and $\cdot$ marks in the literature. The rings do not always have an identity element with respect to multiplication. From now on, we assume that all of our rings are commutative. In the theory of rings, an ideal is a particular subset of a ring. Let $R$ be a ring and assume that $I$ is a subset of $R$. $I$ is said to be an ideal of $R$ if $(I, +)$ is an abelian group and for all $a \in I$ and $r \in R$, we have $ra = ar \in I$. In this work, the definition of ideal in its general form will not be used and we will just use the ideals generated by some elements in $R$. Suppose that $R$ is an arbitrary ring and $\{a_1, a_2, \ldots, a_n\}$ are some elements of it. The ideal generated by $a_1, a_2, \ldots, a_n$ is defined as the smallest ideal (with respect to inclusion) of $R$ containing these elements. This generated ideal also consists of elements in the following form:

$$r_1a_1 + r_2a_2 + \cdots + r_ka_k,$$

where $r_i$'s are elements of $R$. The ideal generated by a single element is called a principal ideal, and the single element is referred to as the generator of ideal. The principal ideal generated by $a$ is represented by $\langle a \rangle$. It is obvious that elements of this ideal are multiples of $a$. Some of the rings have the property that all of their ideals are principal. We call these rings principal ideal rings. We will become acquainted with a set of these rings in the following. A specific instance of rings becomes a key algebraic notion that is directly related to the topic of this research. If the ring $R$ is such that both $(R, +)$ and $(R - \{0\}, \cdot)$ are abelian groups, then $R$ is said to be a field. Immediately following the definition, one can see that every non-zero element has inverse with respect to the second operator, while this is not always the case in ordinary rings. It is obvious that the set of rational numbers, real numbers and complex numbers are fields with their ordinary addition and multiplication operations. It is also proven that $\mathbb{Z}_n$ is a field if and only if $n$ is a prime number. We would be utilizing the $\mathbb{Z}_2$ field extensively in this work.

As we will see in the next section, if $n$ is a prime number, then cyclic codes become basically vector spaces over $\mathbb{Z}_n$. So let us introduce vector spaces over any arbitrary field $\mathbb{F}$. Let $(V, +)$ be an abelian group and let $\mathbb{F}$ be a field. If there exists a function (usually called scalar multiplication) $\cdot : \mathbb{F} \times V \to V$ that meets the following criteria, then $V$ is called a vector space over the field $\mathbb{F}$:

1. For every $v \in V$, $1 \cdot v = v$.

2. For every $v \in V$ and scalars $c_1, c_2 \in \mathbb{F}$, $(c_1c_2) \cdot v = c_1 \cdot (c_2 \cdot v)$.

3. For every scalar $c \in \mathbb{F}$ and every $v_1, v_2 \in V$, $c \cdot (v_1 + v_2) = c \cdot v_1 + c \cdot v_2$.

4. For every $v \in V$ and scalars $c_1, c_2 \in \mathbb{F}$, $(c_1 + c_2) \cdot v = c_1 \cdot v + c_2 \cdot v$.

Generally the elements of vector spaces are called vectors. The concept of subspaces can be defined similarly as subgroups. Let $V$ be a vector space over the field $\mathbb{F}$ and let $W$ be its subset. Then $W$ is said to be the subspace of $V$ if $W$ is a vector space over $\mathbb{F}$ with respect to addition and scalar multiplication of $V$.

**Definition 2.4.** Let $V$ be a vector space over a field $\mathbb{F}$ and let $S = \{v_1, \dots, v_n\}$ be a subset of $V$. $S$ is called linearly independent if every linear combination

$$c_1v_1 + c_2v_2 + \cdots + c_nv_n = 0,$$

where $\forall i, c_i \in \mathbb{F}$, leads us to $c_1 = c_2 = \cdots = c_n = 0$. Otherwise $S$ is called linearly dependent or simply dependent.

According to the preceding definition, as examples, the zero vector (or any set containing the zero vector) is dependent. Every non-zero vector is linearly independent and two vectors are linearly independent if and only if they are not parallel.

Let $V$ be a vector space over the field $\mathbb{F}$ and suppose that $S$ is a set of vectors of $V$. The smallest subspace of $V$ containing $S$ is called the subspace generated by $S$. Similarly, the subspace generated by $S$ is equal to the intersection of all subspaces of $V$ that contain $S$. It is easily proven that elements of the subspace generated by $S$ are of the form $c_1v_1 + c_2v_2 + \cdots + c_kv_k$ where $c_i$'s belong to $\mathbb{F}$. One of the most essential notions in a vector space is dimension, which is directly utilized in theoretical codes. If $S = \{v_1, \dots, v_n\}$ is an independent subset of $V$ and the subspace generated by $S$ is equal to $V$, then $S$ is called a basis for $V$. In this case we say that the dimension of $V$ is equal to $n$, say $\dim V = n$.

For example, $\mathbb{R}^2$ is a vector space of dimension 2 over $\mathbb{R}$. One basis for this space is $\{(1, 0), (0, 1)\}$. Similarly, $\mathbb{R}^3$ is a vector space of dimension 3 and $\mathbb{R}^n$ is the vector space of dimension $n$. It is important to note that when a vector space (even finite dimensional) is studied over an infinite field, one can not have any control on its elements. However, if we have a finite-dimensional vector space over a finite field, then the number of vectors of this vector space equals the number of elements of the field power to the dimension of vector space. Particularly, if you regard $V$ as a 3-dimensional vector space over the field $\mathbb{Z}_2$, this space will then have a total of $2^3 = 8$ vectors, which are as follows:

$$\{(0, 0, 0), (1, 0, 0), (0, 1, 0), (0, 0, 1), (1, 1, 0), (1, 0, 1), (0, 1, 1), (1, 1, 1)\}.$$

This set of vectors can be written as

$$\{000, 100, 010, 001, 110, 101, 011, 111\}.$$

According to engineering disciplines, especially geomatics engineering, the information to be transmitted is often presented as a series of zeroes and ones. A digit is defined as a 0 or a 1. A word is a sequence of these numbers. The number of digits in a word determines its length.

As a result, 0110101 is a seven-letter word. A word is sent via a binary channel by transmitting its digits one after the other. The word “binary" refers to the usage of only two numbers, 0 and 1. Each number is conveyed physically, electronically, magnetically, or otherwise by one of two distinct types of pulses. A binary code $C$ is a collection of words.

For example, the code of all words of length two is equal to

$$C = \{00, 10, 01, 11\}.$$

We can have another type of representation of codewords as polynomials. First of all let us recall what a polynomial is. Let $\mathbb{F}$ be a field.

A polynomial has the representation

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0,$$

where $x$ is an independent variable and $a_i$'s are elements of the field $\mathbb{F}$. $n$ is considered as a non-negative integer. The largest power of $x$ in a polynomial $g(x)$ with a non-zero coefficient is known as the degree of the polynomial and is represented by $\deg(g(x))$. The field which the coefficients of polynomial come from, is very important. If $f(x)$ and $g(x)$ are two polynomials with coefficients in $\mathbb{F}$, then the following equations for the degrees of polynomials are established:

$$\begin{aligned} \deg(f(x) + g(x)) &\leq \max\{\deg(f(x)), \deg(g(x))\}, \\ \deg(f(x)g(x)) &= \deg(f(x)) + \deg(g(x)). \end{aligned}$$

Notice that if $\mathbb{F}$ is not a field, then the last equation may fail. Based on the introductory materials given in the case of the rings, it can be observed that the set of all polynomials with coefficients belonging to the field $\mathbb{F}$ with ordinary addition and multiplication makes a new ring that is called polynomial ring. The polynomial ring with coefficients in the field $\mathbb{F}$ is represented by the symbol $\mathbb{F}[x]$. For example $\mathbb{R}[x]$ consists of all polynomials having real coefficients and $\mathbb{Z}_2[x]$ is the set of all polynomials with coefficients zero and one.

In coding theory, these kinds of rings are employed. Polynomial rings are also known as rings, which are principal ideal rings (in abstract algebra these rings are domains and they are said to be principal ideal domains). In other words, if $I$ is an ideal of $\mathbb{F}[x]$, then there exists the polynomial $g(x)$ such that $I$ is equal to the ideal generated by $g(x)$. This polynomial, which is the generator of a principal ideal, is employed as a cyclic code polynomial generator. This is accomplished easily by applying the division algorithm:

**Theorem 2.5.** *If $f(x)$ and $g(x)$ are polynomials of the ring $F[x]$ with $g(x) \neq 0$, then there are polynomials $q(x)$ and $r(x)$ such that $f(x) = g(x)q(x) + r(x)$ with the property $r(x) = 0$ or $0 \leq \deg(r(x)) < \deg(g(x))$.*

In this division the quotient is $q(x)$, while the remainder of the division is $r(x)$. The remainder of a division in a polynomial ring over a field is uniquely determined. Let $\mathbb{F}[x]$ be the ring of polynomials with coefficients in $\mathbb{F}$, and let $\langle g(x) \rangle$ be the principal ideal generated by $g(x)$. Consider the set

$$f(x) + \langle g(x) \rangle = \{f(x) + h(x) | h(x) \in \langle g(x) \rangle\}.$$

This set is called a coset. The set of all cosets equipped with addition and multiplication operations

$$(f_1(x) + \langle g(x)\rangle) + (f_2(x) + \langle g(x)\rangle) = ((f_1(x) + f_2(x)) + \langle g(x)\rangle),$$

$$(f_1(x) + \langle g(x)\rangle) \cdot (f_2(x) + \langle g(x)\rangle) = ((f_1(x) \cdot f_2(x)) + \langle g(x)\rangle),$$

is a new a ring known as quotient ring.

One represents this ring by $\frac{\mathbb{F}[x]}{\langle g(x)\rangle}$. It is worth noting that in this ring, the additive identity element is $g(x)$ and $f_1(x) + \langle g(x)\rangle = f_2(x) + \langle g(x)\rangle$ if and only if $g(x)|f_1(x) - f_2(x)$.

Recall the $\mathbb{Z}_n$ ring. It can be easily seen that this ring is a quotient ring of integers and each element of this ring is corresponding to the set of $nk+i$s. In fact, $\mathbb{Z}_n$ corresponds algebraically to the quotient ring of $\mathbb{Z}/n\mathbb{Z}$. In the next sections, we will devote a significant amount of attention to this quotient ring.

Let $p$ be a prime number. Then it is easily seen that $\mathbb{Z}_p$ is a (finite) field. If we have an irreducible polynomial over $\mathbb{Z}_p[x]$, say $p(x)$, then the quotient $\mathbb{Z}_p[x]/\langle p(x)\rangle$ is a finite field with $p^m$ elements where $m$ is the degree of $p(x)$. Notice that for every prime number $p$, there exists at least one irreducible polynomial of degree $m$ such that the so called quotient ring is a finite field of order $p^m$. The polynomials over finite fields that permute the elements of the field are the main core of coding theory. These polynomials together with some properties are the generators of codes.

Polynomials, as previously stated, play a vital role in the generation of cyclic codes. Irreducible polynomials and primitive polynomials play a considerably more prominent part in this role.

The polynomial $f(x)$ of the ring $\mathbb{F}[x]$ is said to be irreducible if it cannot be decomposed into two positive degree polynomials. In other words, $f(x)$ is irreducible if and only if the equation $f(x) = g(x)h(x)$ where $0 < \deg(g(x)) < \deg(f(x))$ and $0 < \deg(h(x)) < \deg(f(x))$ is not possible.

Finally, we introduce the primitive polynomials over the $\mathbb{Z}_2$ field (they can be defined over any field). A primitive polynomial is an irreducible polynomial $f(x)$ of degree $m$ in the ring $\mathbb{Z}_2[x]$ such that the lowest positive integer $n$ for which $f(x)$ divides $x^n - 1$ is equal to $2^m - 1$. In other words, the polynomial $f(x)$ of degree $m$ in the ring $\mathbb{Z}_2[x]$ is primitive if it contains a root $a$ such that the set

$$\{0, 1, a, a^2, \ldots, a^{2^m-2}\},$$

equals the Galois field $GF(2^m)$.

What happens with primitive polynomials is that when a number of bits are started as input, the recurrency relation is produced to define the code, which has a length of $2^m-1$ and then these bits are repeated. Indeed, a primitive polynomial generates every LFSR with maximum length. The code length is $2^n - 1$ where $n$ is the length of the input bits for code generation. This is the main reason that the primitive polynomials are used to generate PRN codes.

## 3. Codes

Coding theory is a method for studying the properties of codes and their applications. Codes have numerous applications in different fields and are very important in terms of both information transport and information security.

A block code is a code that has all of its words of the same length; this number is referred to as a code's length. In this note, only block codes will be considered. As a result, for us, the term code will always refer to a binary block code. The words that belong to a certain code $C$ are referred to as codewords. The second point is that detecting the beginning of the first word sent is simple. Thus, if we use codewords of length 3 and obtain 011011001, we know the words are in sequence $011, 011, 001$.

The following statements will demonstrate that the set of elements of a vector space is known as a code. It is observed that these elements may be classified into components, as the result of the addition of two elements is again an element of the same set. Each permutation also gives another vector of the same set.

In general, a code is defined as follows. Consider the Cartesian product of a set $A^n$, where A is a set.

The output of this Cartesian product is known as a block code. In this context, studying this idea in general will be difficult and ineffective. So, because of the scope of this paper, we restrict ourselves to cyclic codes over finite fields in order to demonstrate their functioning and applicability. This is because these codes directly relate to polynomials.

A linear code is one in which the componentwise addition of two items is also one of the elements. More precisely, $C$ is a linear code over the finite field $\mathbb{F}$ with $q$ elements if it is a $k$-dimensional vector subspace. This code is represented by $[n, k]$ over $\mathbb{F}$. As we can see, the code has a completely algebraic structure, and this algebraic structure will have various applications. Every vector space, of course, has a dimension. $C$ (as a vector space) has the dimension $k$ and it has a basis of $k$ vectors that are linearly independent and generate $C$. For this code, a generating matrix of order $k \times n$ is defined, with constituent rows representing the linearly independent elements of the basis of $C$.

Since each code (as an element of a vector space) is a unique linear combination of basis vectors, all elements of this code can be constructed from the introduced generating matrix. Of course, there is another approach to express the code, which is based on calculating the null space of another matrix called the parity check matrix $H$ with order $(n-k) \times n$. In other words, $C$ characterizes the null space of $H$. Using these two matrices, it is simple to determine if an $n$-tuple belongs to $C$ or not (is an element of the code). If $Hc^t = 0$, then $c$ belongs to $C$. It is also feasible to determine the verticality or non-verticality of vector space vectors by utilizing the concept of inner multiplication for the elements of a vector space.

When the inner multiplication of two elements is zero, then they are orthogonal. This definition establishes a link between the generating matrices of $C$ and $C^{\perp}$ codes. This connection states that if $H$ is $C$'s parity check matrix, then $H$ is the generating matrix of $C^{\perp}$. It should be mentioned that the general notion of linear codes can be established on rings, and numerous research has been conducted on them. According to the objectives of this research, we will not discuss codes on rings and will instead focus on codes and their applications on finite fields.

## 3.1 Cyclic Codes

Let $\mathbb{F}$ be a finite field with $k$ elements. An $[n,k]$ code is a sequence of $n$-tuples whose components are from $\mathbb{F}$. Let $v = (v_0, v_1, \ldots, v_{n-1})$ be an $n$-tuple of $\mathbb{F}^n$. Moving these components $i$ places to the right, (In the form of a cyclic code) results in a new $n$-tuple, as

$$v^{(i)} = (v_{n-i}, v_{n-i+1}, \ldots, v_{n-1}, v_0, v_1, \ldots, v_{n-i-1}).$$

This rightward shift by $i$ is the same as the leftward shift by $n-i$. Furthermore, if you consider these components to be a polynomial coefficients, this is equivalent to multiplying the polynomial by $x^i$. These expressions represent a particular state of the cyclic code.

The term "cyclic code" refers to a specific form of linear codes that is widely utilized. If $(c_0, c_1, \ldots, c_{n-1}) \in C$ and one can conclude that $(c_{n-1}, c_0, \ldots, c_{n-2}) \in C$, then we call $C$ a cyclic code. It is self-evident from the definition that $C$ is a cyclic code whenever for every $v \in C$, one can prove that for all $i$, $v^{(i)} \in C$. If in very special case we have $\mathbb{F} = \mathbb{Z}_2$, then $C = \{000, 110, 011, 101\}$ will be a cyclic code; this is extremely simple to explore and this set is essentially a $\mathbb{Z}_2^3$ vector subspace. Clearly, algebra and especially polynomial algebra, over a finite field and cyclic codes are closely related and one can describe these type of codes algebraically.

Keep in mind that each element of $\mathbb{F}^n$ has $n$ components, and if you wish to match this $n$ components with a polynomial, you must limit yourself to polynomials of degree $n-1$. The quotient ring $\frac{\mathbb{F}[x]}{\langle x^n - 1\rangle}$ does this. Consider a one-to-one correspondence

$$\mathbb{F}^n \to \frac{\mathbb{F}[x]}{\langle x^n - 1\rangle},$$

with

$$(c_0, c_1, \ldots, c_{n-1}) \longmapsto c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}.$$

Each element of code $C$ may be represented by a polynomial with coefficients in the field $\mathbb{F}$ and of degree $n-1$ by using this one-to-one correspondence. Because of this relationship, the definition of cyclic codes becomes more algebraic. If $c_0 + c_1 x + \cdots + c_{n-1}x^{n-1} \in C$, the multiplication of $x$ in this expression will be $c_0 x + c_1 x^2 + \cdots + c_{n-1}x^n$. You should divide this phrase by $x^n - 1$ if you wish to view it in $\frac{\mathbb{F}[x]}{\langle x^n - 1\rangle}$. The remainder of this division is

$$c_{n-1} + c_0 x + \cdots + c_{n-2}x^{n-1},$$

which will be a $C$ code element. As previously stated, permuting code digits is equivalent to multiplying $x^i$ by their corresponding polynomial.

As a result, it is possible to say that studying cyclic codes over the field $\mathbb{F}$ is equivalent to studying the ideals of $\frac{\mathbb{F}[x]}{\langle x^n-1\rangle}$. It was shown in the preceding section that the ideals of $\frac{\mathbb{F}[x]}{\langle x^n-1\rangle}$ are all principal ideals and would thus be generated by a polynomial. As a result, each $C$ code is created by a polynomial, which is called as the code's generating polynomial. If $g(x)$ is a generator for code $C$, then you must have $g(x)|x^n-1$; that is, $x^n-1$ must be divisible by the generating polynomial. This is a significant algebraic result for cyclic codes. The following theorem summarizes the above issues.

**Theorem 3.1.** *Assume that $C$ is an $[n,k]$ cyclic code in $\frac{\mathbb{F}[x]}{\langle x^n-1\rangle}$. The monic and unique polynomial $g(x)$ in this instance is such that $\deg(g(x))=n-k$ and $g(x)|x^n-1$. Also $C=\langle g(x)\rangle$.*

If you wish to get the generating matrix of this code, you must first obtain a basis for the required vector space and then arrange the basis elements in the rows of a matrix. Let

$$g(x)=g_0+g_1x+\cdots+g_{n-k}x^{n-k},$$

be the generating polynomial of code $C$ of length $n$ over the finite field $\mathbb{F}$. Then the generating matrix of this code is

$$\begin{bmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0\cdots & \\ 0 & g_0 & g_1 & \cdots & g_{n-k} & 0\cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & \cdots & g_0 & g_1 & \cdots \end{bmatrix}.$$

So the generating matrix of $C$ is the matrix whose rows are multiples of $x^ig(x)$:

$$\begin{bmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-1}g(x) \end{bmatrix}.$$

The parity check matrix can be derived from the notion of quotient-rings. If $g(x)$ is a polynomial that generates the code $C$ over $\mathbb{F}$ with length $n$ and dimension $k$, then $h(x)=(x^n-1)/(g(x))$ is referred to as a parity check polynomial. If $h(x)=h_0+h_1x+\cdots+h_kx^k$, then the ideal generated by $h(x)$ is the same null generator as $C$.

**Example 3.2.** Let $C$ be the code generated by the polynomial $g(x)=1+x+x^3$ on $\mathbb{Z}_2$ with length 7. Then this set is generated by all permutations of $(0,0,0,1,0,1,1)$. Consider the equation

$$h(x)=(x^7-1)/(1+x+x^3)=x^4+x^2+x+1.$$

The rows of generating matrix for this code will be the polynomials in the following order as:

$$g(x), xg(x), x^2g(x), x^3g(x).$$

It is now explained how to construct an $[n, k]$ cyclic code. Suppose that $g(x)$ is a generating polynomial over $\mathbb{Z}_2$ for the code $C$. To code the phrase $(c_0, c_1, \dots, c_{k-1})$ in this situation, first treat it as the polynomial $c(x) = c_0 + c_1x + \cdots + c_{k-1}x^{k-1}$. Then compute $x^{n-k}c(x)$ and divide it by $g(x)$. We know from the division algorithm that there exist $q(x)$ and $r(x)$ such that

$$x^{n-k}c(x) = g(x)q(x) + r(x).$$

Given that the degree of $g(x)$ is equal to $n-k$, the remainder $r(x)$ will have degrees less than or equal to $n-k-1$, according to the division algorithm. The following expression is a multiple of $g(x)$ and hence an element of $C$ because

$$r(x) + x^{n-k}c(x) = q(x)g(x).$$

The code generation process, in general, follows this pattern. In a $[4, 7]$ code with a generating polynomial $g(x) = x^3+x+1$ for example, the approach would be as follows: To code $(0, 1, 1, 0)$, first calculate the relevant polynomials, i.e. multiply $x^2+x$ by $x^3$ and then divide it by $g(x)$. The required code is produced by adding the remainder to the original equation, which in this case is equal to $x^5 + x^4 + 1$, that is equivalent to $(1, 0, 0, 0, 1, 1, 0)$.

Let $C = \langle 1 + x^2\rangle$ with $n = 3$ be over the field $\mathbb{Z}_2$. Elements of $C$ will be the elements of $\langle 1 + x^2\rangle$ in $\frac{\mathbb{Z}_2[x]}{\langle x^2-1\rangle}$. To identify the elements of $C$, just compute all the $r(x)(1 + x^2)$ where $r(x)$ is an element of $\frac{\mathbb{Z}_2[x]}{\langle x^2-1\rangle} := R$. $R$ is made up of the following polynomials:

$$\{0, 1, x, 1 + x, x^2, 1 + x^2, x + x^2, 1 + x + x^2\}.$$

Therefore

$$C = \{0, 1 + x, 1 + x^2, x + x^2\}.$$

That is to say $C = \{000, 110, 101, 011\}$.

This code was introduced as an example of a simple code in the first part. It is worth noting that the number of divisors of $x^n - 1$ can be utilized to produce generating polynomials (and thus code). To compute all 4-length codes on $\mathbb{Z}_3$, for example, first decompose $x^4 - 1$ as

$$x^4 - 1 = (x - 1)(x + 1)(x^2 + 1).$$

It is discovered that $x^4 - 1$ has eight divisors, and each of these divisors will generate a unique code. It is simpler to discover these codes if you know the generating matrix that goes with them. This is only possible if you provide rows of matrices that are multiples of $x^ig(x)$.

Recall that polynomials of degree $m$ are primitive if the least integer $n$ for which $g(x)|x^n - 1$ equals $2^m - 1$. The characteristics of the $\mathbb{Z}_2$ cyclic code generating polynomials may be summarized as follows based on what has been discussed thus far:

1. The degree of $g(x)$ for each $[n, k]$ code must be equal to $n - k = m$.
2. The constant term of $g(x)$ is equal to 1.
3. $g(x)$ is the primitive polynomial of $x^n + 1$.
4. Since we discuss in $\mathbb{Z}_2$, then $x^n + 1 = x^n - 1$.
5. Each element of the code has a maximum degree of $n - 1$.
6. Each code element is a multiple of $g(x)$.

Another advantage of using primitive polynomials to generate codes is that it prevents repetition in the first $2^m - 1$ bits.

## 3.2 Shift Registers

Polynomials can be used to define shift registers, which are utilized for code generation. The code for a given satellite is defined by the initial value of the polynomials and the phase between them. So far, generating polynomials and their types have been recognized.

The linear-feedback shift register (LFSR) idea is now briefly explained. When the $\mathbb{Z}_2$ field is considered, the primitive polynomials can be used to generate quasi-random codes. The primitive polynomials can generate any LFSR with a maximum length of $2^n - 1$, where $n$ is the length of the initial LFSR. In this process, an initial ten-bit code is considered, with each bit allocated a position ranging from 1 to 10. These bits can be thought of as polynomial coefficients on $\mathbb{Z}_2$.

At each step, the bits are now shifted one position to the right. This operation, obviously, corresponds to the multiplication of $x$ in polynomials related to the initial bits. If polynomials like $x^{10} + x^3 + 1$ are used for shifts, then the initial position is determined by adding the third and tenth bits. As a result, $2^{10} - 1$ bits will be generated, and the start attribute of $g(x)$ will prohibit this code from being repeated before $2^{10} - 1$.

This technique is utilized in GNSS satellite systems that send code from satellites to ground receivers to calculate the coordinates of locations. The appropriate polynomial is determined by the number of satellites and does not interact with other satellite systems. Based on the usage of primitive polynomials, the mechanism employed in satellites to transmit information is such that two main polynomials are utilized in such a way that the output of the first polynomial is used as input for the second polynomial.

The stability of displacement, which is an electrical circuit in which the bits are shifted evenly at each stage, performs this operation, and bits are added to the first place of the bits generated. If these adjustments are done one bit at a time (as with satellites), this stability is known as linear displacement. When the polynomial employed is primitive, then the greatest number of bits that could be retrieved without repetition is $2^n - 1$.

# 4. Primitive Polynomials in Geomatics Engineering

All sciences and engineering are founded on mathematics. Furthermore, we may assert unequivocally that mathematics is the structure and engineering is the physical body, [9]. Furthermore, mathematics in engineering focuses on mathematical applications in science and engineering. Geomatics engineering, on the other hand, is a specialized branch of engineering that focuses on the monitoring, implementation, and maintenance of global geospatial information.

Geospatial information is defined as any information that has positional characteristics. Satellite navigation (SATNAV) systems are a subfield of geomatics engineering that employs satellites to offer autonomous geospatial positioning. There are various SATNAV systems in use now all over the world. Some are worldwide, while others solely serve a specific region. The term Global Navigation Satellite System (GNSS) refers to the collection of all SATNAV systems and their enhancements [16].

The GNSS systems are owned by the United States, Global Positioning System (GPS), Chinese BeiDou Navigation Satellite System (BDS), European Galileo, Russian Federation Global Navigation Satellite System (GLONASS), Navigation with Indian Constellation (NavIC), and Japan's Quasi-Zenith Satellite System (QZSS) (Grewal et al., [5]; Winternitz, [27]). The GNSS consists of constellations of Earth orbiting satellites that broadcast their locations in space and time, networks of ground control stations, and receivers that determine ground positions via trilateration.

Figure 1 depicts four widely used GNSS systems. GNSS system positioning is based on a fairly basic idea. To make it function, however, a high level of technological competence is necessary. The satellites transmit an electromagnetic signal, and the user has a receiver that includes a clock. Figure 2 depicts the fundamental principle of GNSS positioning. The user's position may be calculated using the known positions of four satellites $SVN_i$ and the signal travel distance $\rho_i$. A GNSS signal is sent from a satellite to a receiver to determine the distance between the satellite and the receiver.

Each GNSS transmits its signal on a specific radio frequency and constantly transmits signals at two or more frequencies [11]. These signals carry range codes, which allow users to compute the travel time from the satellite to the receiver as well as the satellite coordinates at any time [2, 11]. a ranging code is a series of zeroes and ones that allows the receiver to calculate the time it takes for the radio signal to travel from the satellite to the reception. They are referred to as pseudo-random noise (PRN) sequences or PRN codes [25]. PRN codes contain

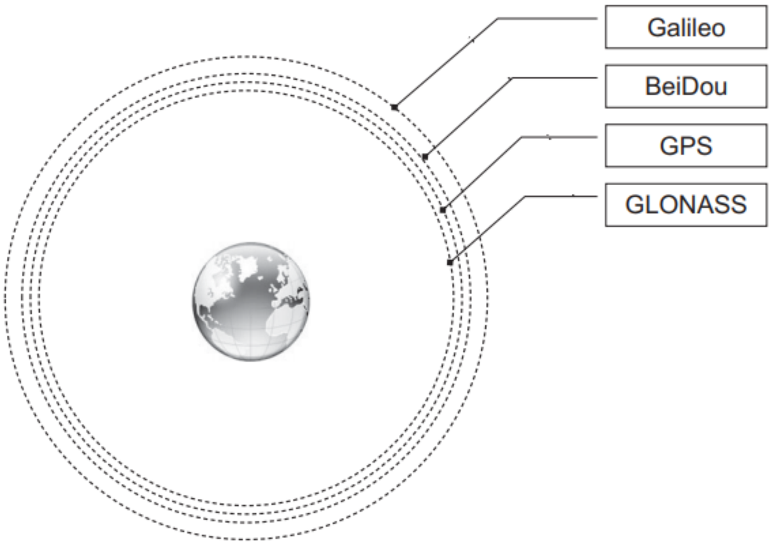


Figure 1: Four well-known GNSS systems are orbiting the Earth [22].

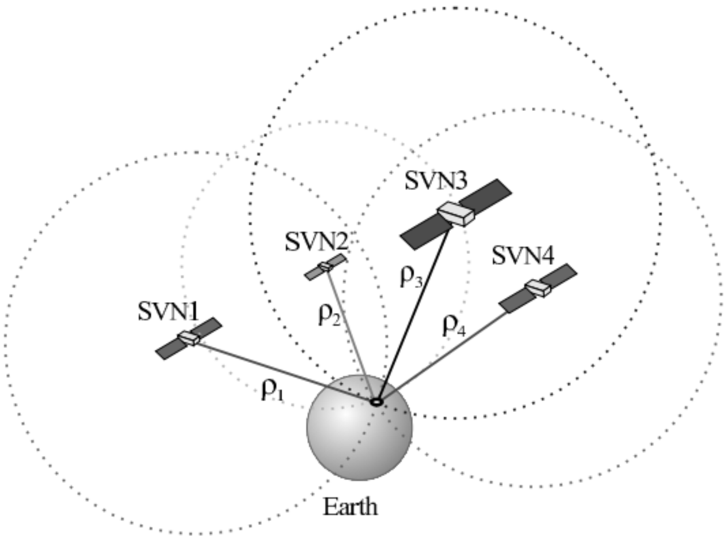


Figure 2: The fundamental idea of GNSS positioning [15, 27].

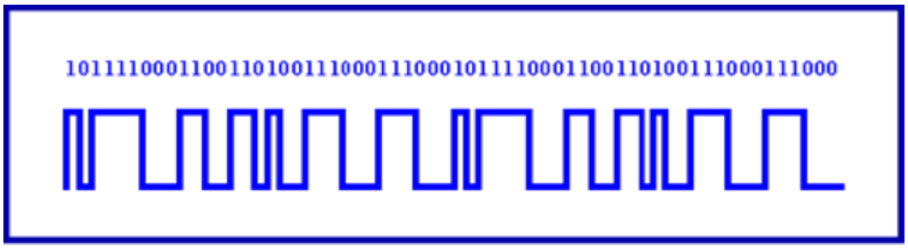


Figure 3: A short repeating PRN code sample [10].

random noise characteristics yet are clearly specified. They are a series of zeroes and ones, with each zero or one referred to as a “chip". Figure 3 illustrates a brief repeating PRN code sample.

The PRN codes have specific mathematical features that allow all satellites to communicate at the same frequency without interfering with one another. These codes also enable exact range measurements between satellites and user devices.

The structure of the PRN code generation is essentially identical for all GNSS systems, however the method of employing mathematical equations in the development of these codes differs somewhat [28]. A deterministic system generates the PRN codes, which are limited in length.

In general, PRN codes are almost orthogonal to one another. In other words, they are nearly completely unrelated. An autocorrelation function with a single correlation peak is also present in the orthogonal codes. The GPS is the first of a new generation of GNSS systems, and it employs two distinct PRN code strings. The Coarse Acquisition Code (C/A-code), often known as the “Civilian Code," and the Precise, or Protected Code (P-Code) are the two [15].

The C/A code is a particular sequence of ones and zeroes that is created 10 times slower than the P-Code. The coding rate for C/A is 1.023 million bits per second. Satellite identification is simple in this case. Each GPS satellite not only broadcasts its own fully unique 1023 bit C/A code, but it also repeats it every millisecond [26]. The C/A code generator, also known as PRN, comprises two shift registers known as Gold polynomial 1 ($G1$) and Gold polynomial 2 ($G2$), which are used to generate Gold code [8].

A Gold sequence, sometimes known as a Gold code, is a type of binary sequence used in communications and GNSS satellites. Robert Gold inspired the name of the Gold codes. Within a set, Gold codes have limited, small cross-correlations, which is beneficial when several devices broadcast in the same frequency band. A Gold code sequence set is made up of $2^n+1$ sequences, each with a period of $2^n-1$, [24]. The procedures below can be used to produce a set of Gold codes. Choose two maximum length sequences of the same length $2^n - 1$ with absolute cross-correlation less than or equal to $2^{(n+2)/2}$, where $n$ is the size of the LFSR used to produce the maximum length sequence. A set of $2^n + 1$ Gold code sequences is formed by the set of the $2^n - 1$ exclusive-OR of the two sequences in their different stages, as well as the two maximum length sequences [8].

In some phases, the exclusive OR of two different Gold codes from the same set equals another Gold code. About half of the codes in a set of Gold codes are balanced -the number of ones and zeroes differs by one. The sequence would include 512 ones and 511 zeroes, which would appear to be randomly dispersed. Furthermore, while Gold codes are not orthogonal, they ensure consistently low cross-correlation with other Gold codes. Because all GPS satellites communicate on the same frequency, this feature is critical.

Gold codes have a maximum length of 1023 chips. To produce signals on all frequencies, the GPS satellite has a 10.23 MHz internal satellite clock. The $L1$ C/A code has a duration of 1 ms. As a result, the chip rate is 1023 chips/ms. Additionally, the Gold codes are generated with the help of a pair of shift registers (gold polynomials) with feedback.

A shift register is just a list of bits with an input and output end. The GPS registers have a length of ten. We'll begin with all 1s, compute the feedback, and then move all of the numbers to the right. In the past sections we explained the meaning of moving the numbers to the right. The GPS C/A range codes are Gold codes from the period $1,023$ [11].

Each satellite has its own code string. The C/A code is the Standard Positioning Service code (SPS). The shift registers are reset to all ones every 1023rd period, causing the function to restart. After the codes are generated, they are merged with the navigation data using Gold polynomial 2 adders. The C/A code generator has two shift registers, $G1$ and $G2$. Each of these shift registers has ten cells that generate sequences of length 1023. The feedback configuration of the $G1$ register is always the polynomial, as shown in the following equation

$$G1: \quad F(X) = 1 + X^3 + X^{10}.$$

This means that states 3 and 10 are returned to the input. Figure 4 depicts the setup of a Gold polynomial ($G1$) shift register generator. Because the sequence repeats itself, just the delay between the polynomials counts in this situation. The delay is determined by selecting certain tap outputs and combining them with an exclusive OR operation. Similarly, the polynomial in the equation

$$G2: \quad F(X) = 1 + X^2 + X^3 + X^6 + X^8 + X^9 + X^{10},$$

applies to the $G2$ register.

Both polynomials starting states are assigned to the same value 1:

$$G1(0) = G2(0) = \{1,1,1,1,1,1,1,1,1,1\}.$$

Figure 5 depicts the setup of a Gold polynomial ($G2$) shift register generator. Thirty-six of the available GPS satellite pairings can be chosen. Figure 6 depicts the GPS satellite combinations used for C/A code generation. As it is seen in

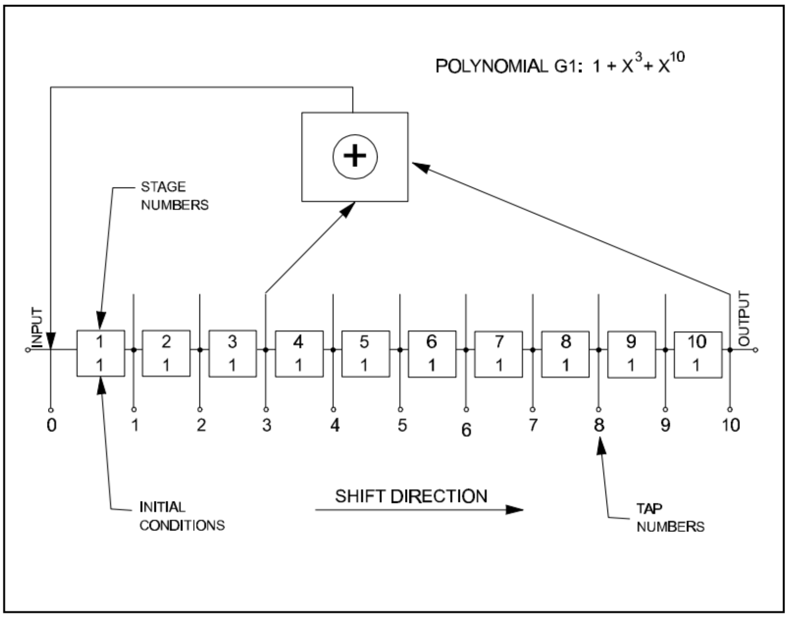


Figure 4: Gold polynomial ($G1$) shift register generator configuration [8].

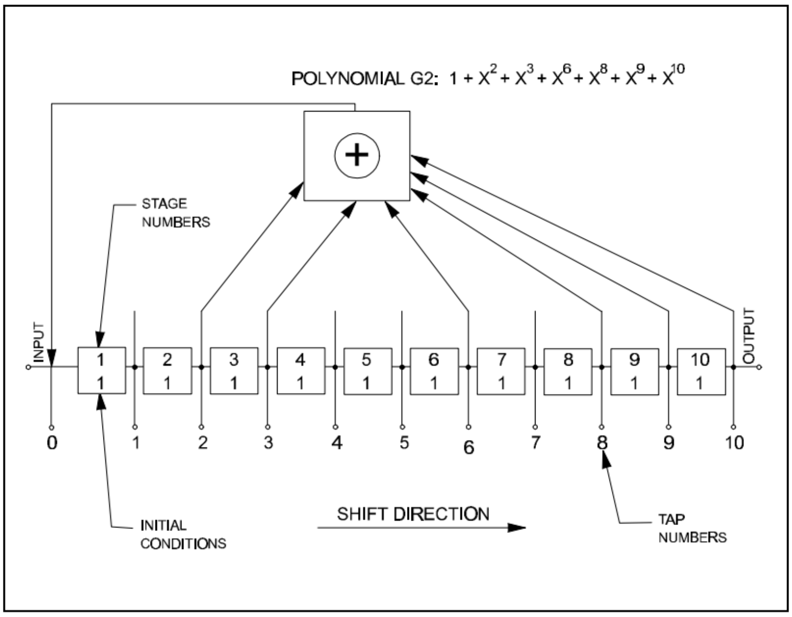


Figure 5: Gold polynomial ($G2$) shift register generator configuration [11].

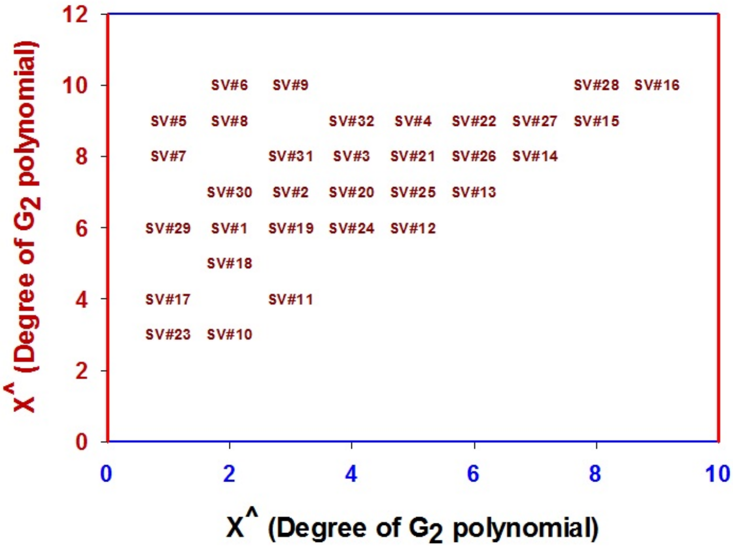


Figure 6: Coordinates of GPS satellites for C/A code generation.

Figure 6, each satellite has a unique combination for generating the C/A code. For example, satellite number 23 employs the power 1 and power 3 of the second polynomial ($G2$), while satellite number 16 employs combinations 9 and 10. A Boolean operator that works on two variables and returns 1 if one but not both of the variables returns 1. So the output of the two shift registers ($G1, G2$) is mixed in an unusual way to generate separate C/A codes for the satellites. The $G1$ register always provides its output, while the $G2$ register generates its output by supplying two of its states to a modulo 2 adder. Phase selection refers to the process of selecting states for the modulo 2 adder.

Moreover, a shift register is a collection of one-bit storage or memory cells. When the register receives a clock pulse, the content of each cell changes one bit to the right. As output, the final cell's content is "read out". The unique characteristics of such shift registers are determined by how information is "read in" to cell 1. The condition of the other cells determines the input to cell 1 in a tapped linear feedback shift register. The binary sum of cells 3 and 10 in a 10-cell register, for example, could be used as the input.

If cells 3 and 10 have distinct states (one is 1 and the other is 0), the following clock pulse will read a 1 into cell 1. If cells 3 and 10 both have the same state, the value 0 is read into cell 1. If we start with 1 in each cell, the contents will be 0010001110 after 12 clock pulses. The next clock pulse will take the 1 in cell 3 and the 0 in cell 10 and add them together to get a total of 1 in cell 1. Meanwhile, all of the other bits have shifted one cell to the right, and the 0 in cell 10 has become

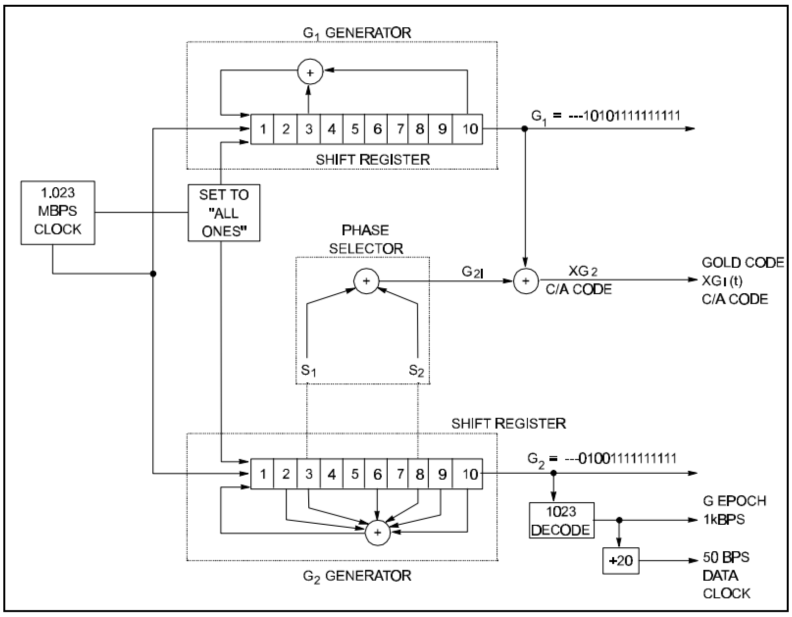


Figure 7: C/A code generation ([14]).

the next bit in the output. Cells $2, 3, 6, 8, 9$ and 10 of the other $G2$ polynomial are tapped and binary-added to produce the new input to cell 1. In this example, the output comes from a different set of taps rather than cell 10.

These second taps are binary-added in various pairs. The various couples produce the same sequence with varied delays or shifts. $G1$'s output is binary-added with the delayed form of the $G2$ sequence. There are 37 PRN C/A codes in all, however two of them (34 and 37) are identical. When old satellites expire and new satellites are launched, a portion of the first 32 codes is allocated to (no more than 24) spacecraft and recycled. Codes 33˘37 are allocated for various uses, such as ground transmitters. Each of the first 32 segments corresponds to a distinct satellite.

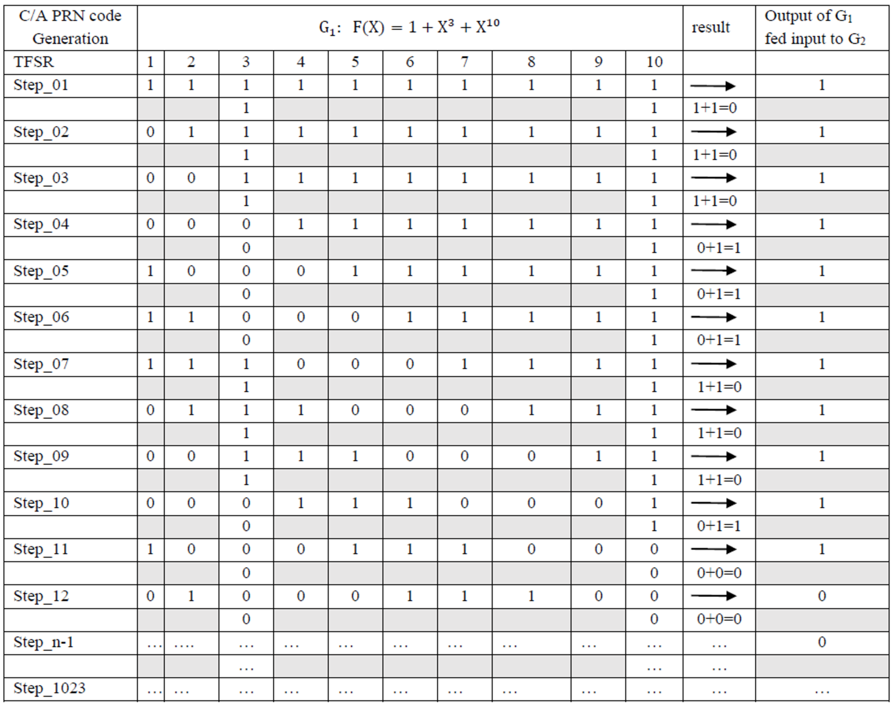
The P code is generated using the same concepts as the C/A code, with the exception that four shift registers with 12 cells are utilized. Modulo 2 addition is used to mix the code with the binary navigation data. If the code chip and the data bits are the same (both 0s and both 1s), the result is 0; otherwise, the result is 1. Modulation is the technique by which the composite binary signal is imposed upon the carrier. Figure 7 depicts the combination of two polynomials as well as the generation of a C/A code. To be more precise, let us explain why the output is 1 in step 11. The input to cell 1 of a tapped LSFR is governed by the state of the other cells. The binary sum of cells 3 and 10 in a 10-cell register $G1$, will be used as the input. If cells 3 and 10 have opposite states (one is 1 and the other is 0), the following clock pulse will read a 1 into cell 1. If cells 3 and 10 both have the same state, the value 0 is read into cell 1.

## 4.3 C/A or PRN Code Generation Steps

As previously stated, the C/A code generator has two shift registers known as $G1$ and $G2$. Each of these shift registers has ten cells that generate sequences of length 1023. The $G1$ register is always in feedback mode with the polynomial as shown. Table 1 shows that the whole sequence of $1,023$ chips is repeated $1,000$ times per second, yielding a “Chip-Rate" of 1.023 MHz or one phase switch (chip) per one-millionth of a second. The shift registers are reset with all ones every 1023rd period, causing the function to restart.

After the codes are generated, they are merged with the navigation data using Gold polynomial 2 adders.

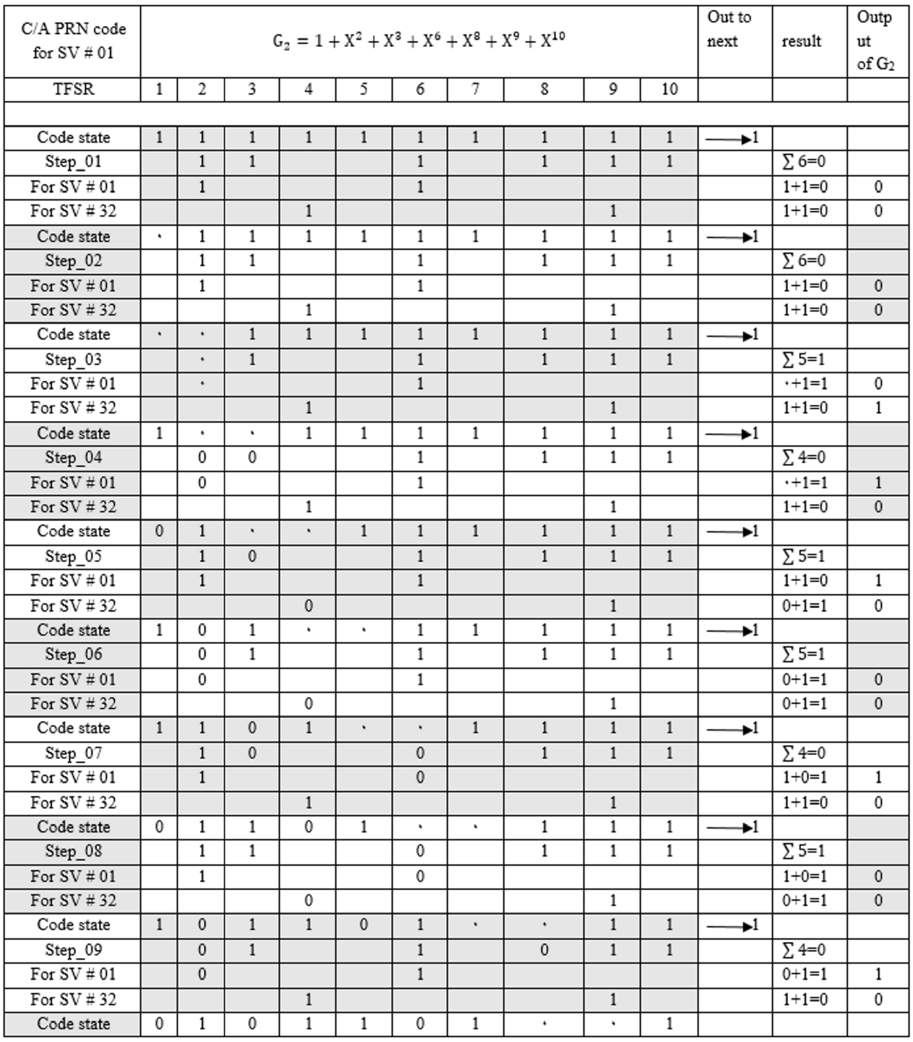
Table 1: Code generation from $G1$.



| C/A PRN code Generation | $G_1$: $F(X) = 1 + X^3 + X^{10}$ | | | | | | | | | | result | Output of $G_1$ fed input to $G_2$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TFSR | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | |
| Step_01 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | ⟶ | 1 |
| | | | 1 | | | | | | | 1 | 1+1=0 | |
| Step_02 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | ⟶ | 1 |
| | | | 1 | | | | | | | 1 | 1+1=0 | |
| Step_03 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | ⟶ | 1 |
| | | | 1 | | | | | | | 1 | 1+1=0 | |
| Step_04 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | ⟶ | 1 |
| | | | 0 | | | | | | | 1 | 0+1=1 | |
| Step_05 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | ⟶ | 1 |
| | | | 0 | | | | | | | 1 | 0+1=1 | |
| Step_06 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | ⟶ | 1 |
| | | | 0 | | | | | | | 1 | 0+1=1 | |
| Step_07 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | ⟶ | 1 |
| | | | 1 | | | | | | | 1 | 1+1=0 | |
| Step_08 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | ⟶ | 1 |
| | | | 1 | | | | | | | 1 | 1+1=0 | |
| Step_09 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | ⟶ | 1 |
| | | | 1 | | | | | | | 1 | 1+1=0 | |
| Step_10 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | ⟶ | 1 |
| | | | 0 | | | | | | | 1 | 0+1=1 | |
| Step_11 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | ⟶ | 1 |
| | | | 0 | | | | | | | 0 | 0+0=0 | |
| Step_12 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | ⟶ | 0 |
| | | | 0 | | | | | | | 0 | 0+0=0 | |
| Step_n-1 | … | …. | … | … | … | … | … | … | … | … | … | 0 |
| | | | … | | | | | | | … | … | |
| Step_1023 | … | … | … | … | … | … | … | … | … | … | … | … |

Similarly, the second table demonstrates how to combine the codes from the $G1$ polynomial and produce PRN codes for each satellite separately using $G2$. This table only includes the first GPS satellite.

Table 2: Code generation using $G2$ only for SV#01 and SV#32.



| C/A PRN code for SV # 01 | $G_2 = 1 + X^2 + X^3 + X^6 + X^8 + X^9 + X^{10}$ | | | | | | | | | | Out to next | result | Output of $G_2$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TFSR | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | | |
| | | | | | | | | | | | | | |
| Code state | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | →1 | | |
| Step_01 | | 1 | 1 | | | 1 | | 1 | 1 | 1 | | ∑ 6=0 | |
| For SV # 01 | | 1 | | | | 1 | | | | | | 1+1=0 | 0 |
| For SV # 32 | | | | 1 | | | | | 1 | | | 1+1=0 | 0 |
| Code state | · | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | →1 | | |
| Step_02 | | 1 | 1 | | | 1 | | 1 | 1 | 1 | | ∑ 6=0 | |
| For SV # 01 | | 1 | | | | 1 | | | | | | 1+1=0 | 0 |
| For SV # 32 | | | | 1 | | | | | 1 | | | 1+1=0 | 0 |
| Code state | · | · | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | →1 | | |
| Step_03 | | · | 1 | | | 1 | | 1 | 1 | 1 | | ∑ 5=1 | |
| For SV # 01 | | · | | | | 1 | | | | | | ·+1=1 | 0 |
| For SV # 32 | | | | 1 | | | | | 1 | | | 1+1=0 | 1 |
| Code state | 1 | · | · | 1 | 1 | 1 | 1 | 1 | 1 | 1 | →1 | | |
| Step_04 | | 0 | 0 | | | 1 | | 1 | 1 | 1 | | ∑ 4=0 | |
| For SV # 01 | | 0 | | | | 1 | | | | | | ·+1=1 | 1 |
| For SV # 32 | | | | 1 | | | | | 1 | | | 1+1=0 | 0 |
| Code state | 0 | 1 | · | · | 1 | 1 | 1 | 1 | 1 | 1 | →1 | | |
| Step_05 | | 1 | 0 | | | 1 | | 1 | 1 | 1 | | ∑ 5=1 | |
| For SV # 01 | | 1 | | | | 1 | | | | | | 1+1=0 | 1 |
| For SV # 32 | | | | 0 | | | | | 1 | | | 0+1=1 | 0 |
| Code state | 1 | 0 | 1 | · | · | 1 | 1 | 1 | 1 | 1 | →1 | | |
| Step_06 | | 0 | 1 | | | 1 | | 1 | 1 | 1 | | ∑ 5=1 | |
| For SV # 01 | | 0 | | | | 1 | | | | | | 0+1=1 | 0 |
| For SV # 32 | | | | 0 | | | | | 1 | | | 0+1=1 | 0 |
| Code state | 1 | 1 | 0 | 1 | · | · | 1 | 1 | 1 | 1 | →1 | | |
| Step_07 | | 1 | 0 | | | 0 | | 1 | 1 | 1 | | ∑ 4=0 | |
| For SV # 01 | | 1 | | | | 0 | | | | | | 1+0=1 | 1 |
| For SV # 32 | | | | 1 | | | | | 1 | | | 1+1=0 | 0 |
| Code state | 0 | 1 | 1 | 0 | 1 | · | · | 1 | 1 | 1 | →1 | | |
| Step_08 | | 1 | 1 | | | 0 | | 1 | 1 | 1 | | ∑ 5=1 | |
| For SV # 01 | | 1 | | | | 0 | | | | | | 1+0=1 | 0 |
| For SV # 32 | | | | 0 | | | | | 1 | | | 0+1=1 | 0 |
| Code state | 1 | 0 | 1 | 1 | 0 | 1 | · | · | 1 | 1 | →1 | | |
| Step_09 | | 0 | 1 | | | 1 | | 0 | 1 | 1 | | ∑ 4=0 | |
| For SV # 01 | | 0 | | | | 1 | | | | | | 0+1=1 | 1 |
| For SV # 32 | | | | 1 | | | | | 1 | | | 1+1=0 | 0 |
| Code state | 0 | 1 | 0 | 1 | 1 | 0 | 1 | · | · | 1 | | | |

The output of the $G1$ polynomial code is passed into the second $G2$ polynomial as an input. Furthermore, Figure 6 depicts GPS satellite combinations for C/A code generation, demonstrating that each satellite has a unique combination for

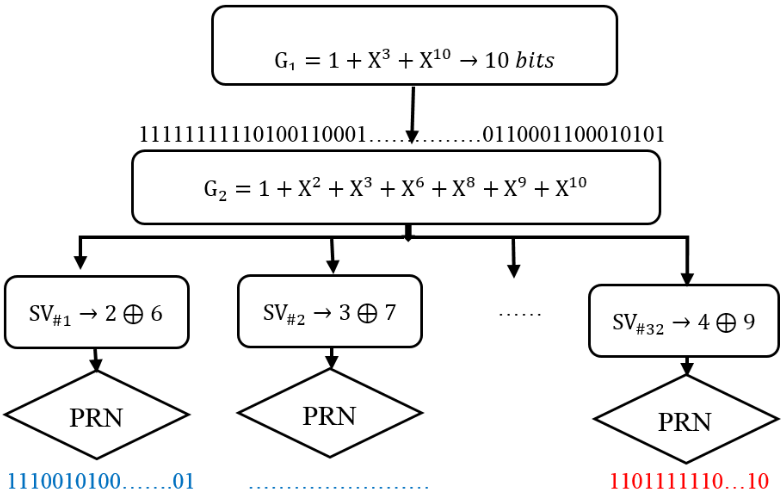
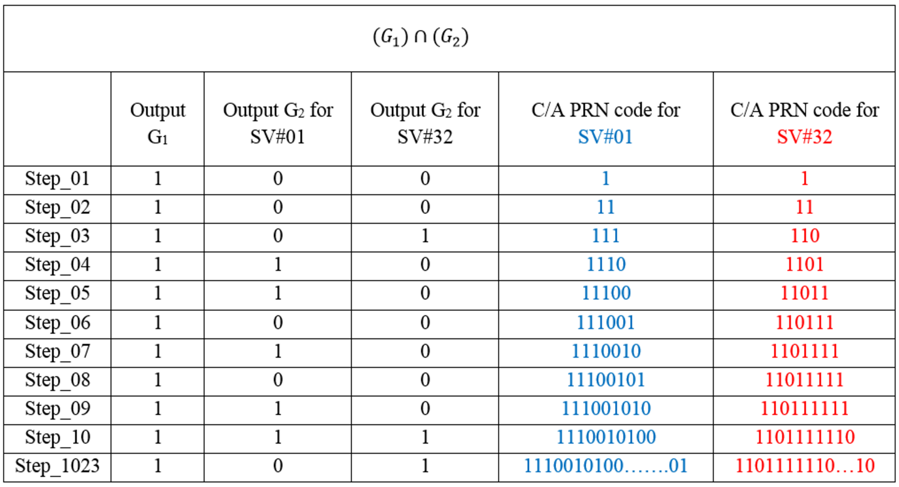

$G_1 = 1 + X^3 + X^{10} \rightarrow 10\ bits$
11111111110100110001..............0110001100010101
$G_2 = 1 + X^2 + X^3 + X^6 + X^8 + X^9 + X^{10}$
$SV_{\#1} \rightarrow 2 \oplus 6$
$SV_{\#2} \rightarrow 3 \oplus 7$
......
$SV_{\#32} \rightarrow 4 \oplus 9$
PRN
PRN
PRN
1110010100.......01
.........................
1101111110...10

Figure 8: The combinations of the $G1$ and $G2$ outputs to generate a unique PRN or C/A code for each GPS satellite.

generating the C/A code. Figure 8 depicts the $G1$ and $G2$ polynomial output combinations used to produce a unique code for each satellite.

Table 3: C/A PRN Code generation for SV#01 and SV#32.



| $(G_1) \cap (G_2)$ | | | | | |
|---|---|---|---|---|---|
| | Output $G_1$ | Output $G_2$ for SV#01 | Output $G_2$ for SV#32 | C/A PRN code for SV#01 | C/A PRN code for SV#32 |
| Step_01 | 1 | 0 | 0 | 1 | 1 |
| Step_02 | 1 | 0 | 0 | 11 | 11 |
| Step_03 | 1 | 0 | 1 | 111 | 110 |
| Step_04 | 1 | 1 | 0 | 1110 | 1101 |
| Step_05 | 1 | 1 | 0 | 11100 | 11011 |
| Step_06 | 1 | 0 | 0 | 111001 | 110111 |
| Step_07 | 1 | 1 | 0 | 1110010 | 1101111 |
| Step_08 | 1 | 0 | 0 | 11100101 | 11011111 |
| Step_09 | 1 | 1 | 0 | 111001010 | 110111111 |
| Step_10 | 1 | 1 | 1 | 1110010100 | 1101111110 |
| Step_1023 | 1 | 0 | 1 | 1110010100.......01 | 1101111110...10 |

As indicated in the above table, the C/A PRN codes for satellites No. 1 and 32 GPS are produced in 10 states that are distinct and unique to each satellite. For each GNSS satellite, these codes are generated and repeated into 1023 strings of PRN code every microsecond. Following that, the GPS carrier's waves may have been modulated in a variety of ways to convey the binary codes, which are 0s and 1s. In any case, binary phase shift keying is the most widely utilized spread spectrum modulation method (BPSK). This method was used to generate the NAV Message, the $P(Y)$ code, and the C/A code.

Binary biphasic modulation is the process of transitioning from 0 to 1 and 1 to 0 by 180 phase shifts in the carrier wave. PRN codes are made up of a series of zeroes and ones, with each zero or one referred to as a "chip". These codes are concatenated and modulated with navigation data before being transmitted to the ground.

### 4.4 Gold Polynomial Coding for Additional GNSS Systems

The structure of PRN code generation is essentially identical in all GNSS systems, however the method of employing mathematical equations in the development of these codes differs somewhat. The following equations show the gold code generating polynomials for BeiDou satellite systems [28]:

$$G1(X) = 1 + x + x^7 + X^8 + X^9 + X^{10} + X^{11},$$
$$G2(X) = 1 + x + x^2 + X^3 + X^4 + X^5 + X^8 + X^9 + X^{11},$$

with initial states

$$G1(0) = G2(0) = \{0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0\}.$$

The range codes are different from GPS, even for the same taps, because the polynomials and their initial states are different.

Similarly, the gold code generating polynomial for GLONASS satellite systems is represented by equations

$$G1(X) = 1 + X^5 + X^9,$$
$$G2(X) = 1 + X^3 + X^5.$$

## 5. Conclusion

Coding theory, sometimes known as algebraic coding theory, is the study of strategies for efficiently and accurately transmitting information from one location to another. This research not only presents and discusses the generation of PRN codes for GNNS satellites using various polynomials, but it also illustrates a practical application of mathematics in geomatics engineering. In recent decades, algebraic geometry has found fascinating applications in coding theory and cryptography.

To this end, we investigated the application of the polynomial ring on finite fields in the generation of various satellite codes and their transmission to Earth to establish coordinates and other purposes.

This paper discussed advanced elements of coding and cryptography in the Global Navigation Satellite System (GNSS). In GNSS, satellites continually broadcast signals at two or more frequencies, whose transmissions include pseudo-random noise (PRN) codes.

Each GNSS satellite has its own unique PRN code, which allows all satellites to communicate at the same frequency without interfering with one another due to the unique mathematical features of PRN codes. The PRN code generator employs two shift registers known as gold polynomials.

The number of satellites and their interacts with other satellite systems decide the suitable polynomial. The technique used in satellites to transmit information is based on the use of primitive polynomials, in which two primitive polynomials are used in such a way that the output of the first polynomial is used as input for the second polynomial. This process is performed by the stability of displacement, which is an electrical circuit in which the bits are displaced equally at each stage, and bits are added to the first of the number of bits generated. This stability is known as linear displacement when these changes are made one bit at a time (as with satellites).

When the used polynomial is primitive, the most bits that can be recovered without repetition is $2^n - 1$. The PRN codes have unique mathematical features that allow all satellites to communicate at the same frequency without interfering with one another, as well as exact range measurements between satellites and user devices. The structure of the PRN code generation utilizing polynomials is essentially identical to all GNSS systems, however the method of applying mathematical equations in the development of these codes is slightly different.

The PRN code generator has two shift registers and is constantly in feedback mode with the polynomial equations. They are configured by selecting particular tap outputs and merging them with an exclusive OR operation. Polynomials can be used to define shift registers, which are utilized for code generation. The correlation measurements define the basic properties of the PRN sequences. Two infinite random sequences should be unrelated to one another. This study presents and discusses the generation of PRN codes for GNNS satellites using various polynomials.

**Acknowledgements.** The authors would like to thank the reviewers for their valuable suggestions and comments.

**Conflicts of Interest.** The authors declare that there are no conflicts of interest regarding the publication of this article.

## References

[1] F. Arnault, T. Berger, M. Minier and B. Pousse, Revisiting LFSRs for cryptographic applications, *IEEE Trans. Inf. Theory* **57** (12) (2011) $8095 - 8113$.

[2] J. L. Awange and J. B. Kyalo Kiema, Modernization of GNSS, In: Environmental Geoinformatics, Environmental Science and Engineering, Springer, Berlin, Heidelberg (2013) pp. $47 - 54$.

[3] R. Chaudhary and V. Gupta, Error control techniques and their applications, *Int. J. Comput. Appl. Eng. Sci.* **I** (II) (2011) $187 - 191$.

[4] C. Flaut, Some application of difference equations in cryptography and coding theory, *J. Difference Equ. Appl.* **25** (7) (2019) $905 - 920$.

[5] M. S. Grewal, A. P. Andrews and C. G. Bartone, Global Navigation Satellite Systems, Inertial Navigation, and Integration, 4th ed., John Wiley & Sons Inc., Hoboken, NJ, 2020.

[6] G. N. Gulak, Method for constructing primitive polynomials for cryptographic subsystems of dependable automated systems, *J. Automation Inf. Sci.* **52** (12) (2020) $52 - 57$.

[7] W. Guo and F. W. Fu, Semilinear transformations in coding theory and their application to cryptography, *arXiv* : 2107.03157 (2020).

[8] M. M. Hafidhi, E. Boutillon and C. Winstead, Reliable gold code generators for GPS receivers, In: IEEE 58th Int. Midwest Symposium on Circuits and Systems (MWSCAS), ort Collins, CO (2015) pp. $1 - 4$.

[9] D. Harris, L. Black, P. Hernandez-Martinez, B. Pepin, J. Williams and W. T. T. Team, Mathematics and its value for engineering students: what are the implications for teaching?, *Int. J. Math. Edu. Sci. Tech.* **46** (3) (2015) $321 - 336$.

[10] C. J. Hegarty, GNSS signals-An overview, In: IEEE Int. Frequency Control Symposium Proc., Baltimore, MD, USA (2012) pp. $1 - 7$.

[11] C. J. Hegarty, A simple model for GPS C/A-code self-interference, *J. Institute Navigation* **67** (2) (2020) $319 - 331$.

[12] M. Hell, Using coding techniques to analyze weak feedback polynomials, In: IEEE Int. Symposium Inf. Theory, Austin, TX, USA (2010) pp. $2523 - 2527$.

[13] G. I. Ivchenko, Y. I. Medvedev and V. A. Mironova, Krawtchouk polynomials and their applications in cryptography and coding theory, *Mat. Vopr. Kriptogr.* **6** (1) (2015) $33 - 56$.

[14] B. P. Kumar and C. S. Paidimarry, Development and analysis of C/A code generation of GPS receiver in FPGA and DSP, In: Proc. Conf. Recent Adv. Eng. Comput. Sci. (RAECS), Chandigarh, India (2014) pp. 1 – 5.

[15] R. B. Langley, P. J. Teunissen and O. Montenbruck, *Introduction to GNSS,* In: P. J. Teunissen and O. Montenbruck (eds) Springer Handbook of Global Navigation Satellite Systems, Springer Handbooks, Springer, Cham, 2017.

[16] W. Lechner and S. Baumann, Global navigation satellite systems, *Comput. Elect. Agriculture* **25** (1-2) (2000) 67 – 85.

[17] J. L. Massey, A short introduction to coding theory and practice, In: Proc. Int. Symp. on Signals, Systems and Electronics (ISSSE '89), Erlangen, Germany (1989) pp. 6.29 – 6.33.

[18] S. Mesnager, Semi-bent functions from oval polynomials, In: Stam, M. (ed.), IMA Int. Conf. Cryptography and Coding (IMACC 2013), LNCS, Vol. 8308, Springer, Heidelberg (2013) pp. 1 – 15.

[19] I. Muchtadi-Alamsyah, Algebraic structures in cryptography, Proc. Int. Conf. Math. Sci. (Keynote Speaker), 2011.

[20] M. W. Paryasto, B. Rahardjo, I. Muchtadi-Alamsyah and M. Hafiz, Implementation of Polynomial-ONB I Basis Conversion, *J. Ilmiah Teknik Komputer* **i** (2) (2010).

[21] S. Puchinger and A. Wachter-Zeh, Fast operations on linearized polynomials and their applications in coding theory, *J. Symb. Comput.* **89** (2018) 194–215.

[22] C. Shi and N. Wei, *Satellite Navigation for Digital Earth*, In: H. Guo, M. F. Goodchild, A. Annoni, (eds) Manual of Digital Earth, Springer, Singapore, 2020.

[23] D. Torrieri, *Principles of Spread-Spectrum Communication Systems*, Maryland: Springer International Publishing Switzerland, 2015.

[24] H. N. Viet, K. R. Kwon, K. S. Moon and S. H. Lee, Simulation model implementation of GPS IF signal generator, *Int. Conf. Inf. Commun.* (ICIC), Hanoi, Vietnam (2017) 17084069.

[25] S. Wallner, J. -A. Avila-Rodriguez, G. W. Hein and J. J. Rushanan, Galileo E1 OS and GPS L1C pseudo-random noise codes-requirements, generation, optimization and comparison, Proc. 20th Int. Technical Meeting of the Satellite Division of The Institute of Navigation (ION-GNSS 2007) (2007) pp. 1549 – 1563.

[26] W. Guan, Y. Wu, S. Wen, C. Yang, H. Chen, Z. Zhang and Y. Chen, High precision three-dimensional iterative indoor localization algorithm using code division multiple access modulation based on visible light communication, *Optical Eng.* **55** (10) (2016) 106105.

[27] L. Winternitz, Introduction to GPS and other global navigation satellite systems, In: Annual Time and Frequency Metrology Seminar, no. GSFC-E-DAA-TN42241, 2017.

[28] J. Xie, H. Wang, P. Li and Y. Meng, *Satellite Navigation Systems and Technologies*, Springer, Singapore, 2021.

[29] Y. Zheng, A note on a class of permutation polynomials, 4th Int. Conf. Emerging Intelligent Data and Web Technologies (EIDWT) (2013) pp. $302 - 306$.

[30] Y. Zheng, Q. Wang and W. Wei, On inverses of permutation polynomials of small degree over finite fields, *IEEE Trans. Inf. Theory* **66** (2) (2019) $914 - 922$.

Amir Bagheri
Department of Fundamental Sciences
Marand Faculty of Engineering
University of Tabriz
Tabriz, I. R. Iran
e-mail: a_bageri@tabrizu.ac.ir

Hassan Emami
Department of Geomatics Engineering
Marand Faculty of Engineering
University of Tabriz
Tabriz, I. R. Iran
e-mail: h_emami@tabrizu.ac.ir