Design of a Code using System Theoretic Approach

V. C. Kotak[1] and B. K. Lande[2]

[1] Research Scholar, Veermata Jijabai Technological Institute, Matunga, Mumbai, INDIA

[2] School of Engineering & Technology, Himalayan University, Jolly Grant, Dehradun, INDIA

Abstract

To design the coding and decoding schemes which can be easily implemented has been the focus of researchers. We propose here one such scheme which is based on the definition of controllability of linear discrete-time system. Given a linear controllable discrete-time system, it has been proposed here that a good code with $d_{min} = 2$ can be proposed. Such a code has been shown to satisfy Singleton as well as Plot kin bounds.

Keywords: *Discrete-time system, Controllability, Coset leaders, Decoders.*

1. Introduction

In an effort to find good codes for practical purpose, researchers have moved beyond block codes to other paramounts such as convolution codes, turbo codes, space-time codes, low-density parity check codes and even quantum codes.

All the coding and decoding schemes proposed by now suffer from limitations due to the complexity of the decoder; therefore, all efforts in this direction have been focused towards the design of coding and decoding schemes which could be easily implemented.

Consider a linear controllable discrete-time scalar control system

$$x(k+1) = Ax(k) + bu(k) \qquad (1)$$

Controllability of this system implies that it is always possible to design a control function consisting of a sequence of impulse functions that can steer all the states of a given system in (1) to the origin. Such a control strategy that consists of sequence of impulse functions cannot be designed for continuous-time linear controllable system. One technique of designing such a control strategy for continuous-time systems has been proposed [1]. It was therefore thought that by applying similar idea to discrete-time linear controllable systems, the states reach the origin by a suitable action of control. Such control functions have been proposed in [2]. It was then thought that if the same idea is applied to linear discrete time controllable system then all the states at the n^{th} instant of sampling generate a set of n linearly independent vectors. We propose this set as a kind of code and stud of the paper.

We first introduce the concept of steering the states to the origin for a system described in terms of a transfer function of a linear discrete time controllable system.

Let us consider equation (1) which is an n^{th} order difference equation,

$$y(k+n)+C_1 y(k+n-1)+C_2 y(k+n-2)+....+C_n y(k)=u(k) \qquad (2)$$

with the initial conditions,

$$\left[y(0) \quad y(1) \quad - \quad - \quad - \quad y(n-1)\right]^T = \left[d_0 \quad d_1 \quad - \quad - \quad - \quad d_{n-1}\right]^T \qquad (3)$$

and it is required that

$$\left[y(n) \quad y(n+1) \quad - \quad - \quad - \quad y(2n-1)\right]^T = \left[0 \quad 0 \quad - \quad - \quad - \quad 0\right]^T \qquad (4)$$

In order to do so, let us assume that the solution of (2) can be given by

$$y(k) = \sum_{j=0}^{n-1} a_j \partial(k-j) \qquad (5)$$

Using the given initial conditions, we can work out the values of a_j , $j = 0,1,....,n-1$. This leads to,

$$a_0 = d_0, ..., a_{n-1} = d_{n-1} \qquad (6)$$

Looking upon (2) as an expression for u(k), and then substituting for all the difference terms of y(k) in terms of 'δ' functions using (5), we finally get an expression for u(k) , in terms of a sequence of impulse functions that will steer $y(0)$, $y(1)$, ..., $y(n-1)$ to the origin.

This idea can be extended to the discrete-time linear system represented in the state variable form for designing the code.

2. Problem Statement

Consider a system in (1) with the initial conditions

$\left[x(0),\ x(1),\ .....,\ x(n-1)\right]^T \triangleq \left[d_1,\ d_2,\ .....,\ d_n\right]^T$, of which the general solution can be given by,

$$x(n) = A^n x(0) + \sum_{j=0}^{n-1} A^j B u(n-1-j) \qquad (7)$$

Let us assume that $x_1(k)$ which is the k^{th} component of the n-tuple vector x_i to be given by,

$$x_1(k) = c_1\delta(k) + c_2\delta(k-1) + ... + c_n\delta(k-n+1) \qquad (8)$$

where c_i are the constants to be determined using the initial conditions, note that k is a general variable and n denotes the dimension of the system.

Evaluating (8) at $k = 0$, we get $c_1 = x_1(0)$

Further,

$$x_1(k+1) = c_1\delta(k+1) + c_2\delta(k) + c_3\delta(k-1) + ... + c_n\delta(k-n+2) \triangleq x_2(k) \qquad (9)$$

And evaluating (9) at, $k = 0$ leads to $c_2 = x_2(0)$

continuing in the same way, we shall obtain

$$x_2(k+1) = c_1\delta(k+2) + c_2\delta(k+1) + c_3\delta(k) + ... + c_n\delta(k-n+3) \triangleq x_3(k) \qquad (10)$$

leading to, $c_3 = x_3(0)$ and finally,

$$x_{n-1}(k+1) = c_1\delta(k+n-1) + c_2\delta(k+n-2) + c_3\delta(k+n-3) + ... + c_n\delta(k) \triangleq x_n(k) \qquad (11)$$

and thus, $c_n = x_n(0)$ this results in

$$x_1(k) = d_1\delta(k) + d_2\delta(k-1) + ... + d_n\delta(k-n+1) \qquad (12)$$

Since (1) is controllable, we can express matrices A and b in phase variable canonical form, so that (1) can be represented as,

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ | \\ | \\ | \\ x_{n-1}(k+1) \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & - & 0 & 0 & 0 \\ 0 & 0 & 1 & - & 0 & 0 & 0 \\ | & & & | & & & | \\ | & & & | & & & | \\ 0 & 0 & 0 & - & 0 & 1 & 0 \\ 0 & 0 & 0 & - & 0 & 0 & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & - & -a_3 & -a_2 & -a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ | \\ | \\ | \\ x_{n-1}(k) \\ x_n(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ | \\ | \\ | \\ 0 \\ 1 \end{bmatrix} u(k) \qquad (13)$$

Thus,

$$x_1(k+1) = x_2(k)$$
$$x_2(k+1) = x_3(k)$$
$$\vdots$$
$$x_{n-1}(k+1) = x_n(k)$$

and

$$x_n(k+1) = -a_n x_1(k) - a_{n-1}x_2(k) - .... - a_1 x_n(k) + u(k) \qquad (14)$$

or

$$u(k) = x_n(k+1) + a_n x_1(k) + a_{n-1}x_2(k) + ... + a_1 x_n(k) \qquad (15)$$

treating this as the formula for u(k), we replace $x_1(k), x_2(k), ..., x_n(k)$ & $x_n(k+1)$

in terms of impulse functions $'\delta'$ by using the assumed (16). This finally leads us to

$$\begin{aligned} u(k) &= \left[d_1\partial(k+n) + d_2\partial(k+n-1) + ... + d_n\partial(k+1)\right] \\ &\quad + a_n\left[d_1\partial(k) + d_2\partial(k-1) + ... + d_n\partial(k-n+1)\right] \\ &\quad + a_{n-1}\left[d_1\partial(k+1) + d_2\partial(k) + ... + d_n\partial(k-n+2)\right] \\ &\qquad\quad | \qquad\qquad | \qquad\qquad | \\ &\qquad\quad | \qquad\qquad | \qquad\qquad | \\ &\quad + a_1\left[d_1\partial(k+n-1) + d_2\partial(k+n-2) + ... + d_n\partial(k)\right] \end{aligned}$$

thus we get,

$$\begin{aligned} &\therefore\ u(0) = a_n d_1 + a_{n-1}d_2 + a_{n-2}d_3 + ... + a_2 d_{n-1} + a_1 d_n \\ &\therefore\ u(1) = a_n d_2 + a_{n-1}d_3 + ... + a_3 d_{n-1} + a_2 d_n \\ &| \qquad | \qquad | \\ &| \qquad | \qquad | \\ &\therefore\ u(n-1) = a_n d_n \end{aligned} \qquad (16)$$

Now, for $x(1)$ we have

$$x(1) = Ax(0) + bu(0)$$

$$x(1) = \begin{bmatrix} 0 & 1 & 0 & - & 0 & 0 & 0 \\ 0 & 0 & 1 & - & 0 & 0 & 0 \\ | & & & | & & & | \\ | & & & | & & & | \\ 0 & 0 & 0 & - & 0 & 1 & 0 \\ 0 & 0 & 0 & - & 0 & 0 & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & - & -a_3 & -a_2 & -a_1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ | \\ | \\ | \\ d_{n-1} \\ d_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ | \\ | \\ | \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} a_n d_1 & a_{n-1}d_2 & \cdots & a_2 d_{n-1} & a_1 d_n \end{bmatrix}$$

$$x(1)=\begin{bmatrix} d_2 \\ d_3 \\ | \\ | \\ | \\ d_{n-1} \\ -a_n d_1 - a_{n-1}d_2 - \ldots.. - a_2 d_{n-1} - a_1 d_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ | \\ | \\ | \\ 0 \\ a_n d_1 + a_{n-1}d_2 + \ldots.. + a_2 d_{n-1} + a_1 d_n \end{bmatrix}$$

$$x(1)=\begin{bmatrix} d_2 \\ d_3 \\ | \\ | \\ | \\ d_{n-1} \\ -a_n d_1 - a_{n-1}d_2 - \ldots.. - a_2 d_{n-1} - a_1 d_n + a_n d_1 + a_{n-1}d_2 + \ldots.. + a_2 d_{n-1} + a_1 d_n \end{bmatrix}$$

$$x(1)=\begin{bmatrix} d_2 \\ d_3 \\ | \\ | \\ | \\ d_{n-1} \\ 0 \end{bmatrix} \tag{17}$$

further for $x(2)$ we have,

$$x(2) \; = \; A^2 x(0) + bu(1) + Abu(0)$$

$$x(2)=\begin{bmatrix} 0 & 1 & 0 & - & 0 & 0 & 0 \\ 0 & 0 & 1 & - & 0 & 0 & 0 \\ | & & & | & & & | \\ | & & & | & & & | \\ 0 & 0 & 0 & - & 0 & 1 & 0 \\ 0 & 0 & 0 & - & 0 & 0 & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & - & -a_3 & -a_2 & -a_1 \end{bmatrix}^2 \begin{bmatrix} d_1 \\ d_2 \\ | \\ | \\ d_{n-2} \\ d_{n-1} \\ d_n \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ 0 \\ | \\ | \\ | \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} a_n d_2 & a_{n-1}d_3 & - & - & - & a_3 d_{n-1} & a_2 d_n \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 1 & 0 & - & 0 & 0 & 0 \\ 0 & 0 & 1 & - & 0 & 0 & 0 \\ | & & & | & & & | \\ | & & & | & & & | \\ 0 & 0 & 0 & - & 0 & 1 & 0 \\ 0 & 0 & 0 & - & 0 & 0 & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & - & -a_3 & -a_2 & -a_1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ | \\ | \\ | \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} a_n d_1 & a_{n-1}d_2 & \cdots & a_2 d_{n-1} & a_1 d_n \end{bmatrix}$$

$$x(2)=\begin{bmatrix} 0 & 1 & 0 & - & - & - & 0 \\ 0 & 0 & 1 & - & - & - & 0 \\ | & & & | & & & | \\ | & & & | & & & | \\ | & & & | & & & | \\ -a_n & -a_{n-1} & -a_{n-2} & - & - & -a_2 & -a_1 \\ a_n a_1 & -a_n + a_1 a_{n-1} & -a_{n-1} + a_1 a_{n-2} & - & - & -a_3 + a_1 a_2 & -a_2 + a_1^2 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ | \\ | \\ | \\ d_{n-1} \\ d_n \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ 0 \\ | \\ | \\ | \\ 0 \\ a_n d_2 + a_{n-1}d_3 + \ldots.. + a_3 d_{n-1} + a_2 d_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ | \\ | \\ | \\ 1 \\ -a_1 \end{bmatrix} \begin{bmatrix} a_n d_1 & a_{n-1}d_2 & \cdots & a_2 d_{n-1} & a_1 d_n \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ 0 \\ | \\ | \\ | \\ 1 \\ -a_1 \end{bmatrix} \begin{bmatrix} a_n d_1 & a_{n-1}d_2 & a_{n-2}d_3 & - & - & a_2 d_{n-1} & a_1 d_n \end{bmatrix}$$

$$x(2)=\begin{bmatrix} d_2 \\ d_3 \\ | \\ | \\ d_{n-2} \\ -a_n d_1 - a_{n-1}d_2 - a_{n-2}d_3 - \ldots\ldots\ldots\ldots - a_2 d_{n-1} - a_1 d_n \\ a_1 a_n d_1 - a_n d_2 + a_1 a_{n-1} d_2 - \ldots.. - a_3 d_{n-1} + a_1 a_2 d_{n-1} - a_2 d_n + a_1^2 d_n \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ 0 \\ | \\ | \\ | \\ 0 \\ a_n d_2 + a_{n-1}d_3 + \ldots.. + a_3 d_{n-1} + a_2 d_n \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ 0 \\ | \\ | \\ | \\ a_n d_1 + a_{n-1}d_2 + a_{n-2}d_3 + \ldots\ldots\ldots\ldots + a_2 d_{n-1} + a_1 d_n \\ -a_1 a_n d_1 - a_1 a_{n-1} d_2 - a_1 a_{n-2} d_3 - \ldots.. - a_1 a_2 d_{n-1} - a_1^2 d_n \end{bmatrix}$$

$$x(2)=\begin{bmatrix} d_2 \\ d_3 \\ | \\ | \\ d_{n-2} \\ -a_n d_1 - a_{n-1}d_2 - a_{n-2}d_3 - \ldots - a_2 d_{n-1} - a_1 d_n + a_n d_1 + a_{n-1}d_2 + a_{n-2}d_3 + \ldots + a_2 d_{n-1} + a_1 d_n \\ a_1 a_n d_1 - a_n d_2 + a_1 a_{n-1} d_2 - a_{n-1} d_3 + a_1 a_{n-2} d_3 - .. \\ \qquad + a_3 d_{n-1} + a_2 d_n - a_1 a_n d_1 - a_1 a_{n-1} d_2 - a_1 a_{n-2} d_3 - \ldots\ldots - a_1 a_2 d_{n-1} - a_1^2 d_n \end{bmatrix}$$

$$x(2)=\begin{bmatrix} d_2 \\ d_3 \\ | \\ | \\ d_{n-2} \\ 0 \\ 0 \end{bmatrix} \tag{18}$$

Continuing the same process to obtain the values of $x(3),\ x(4),.....,x(n)$, we observe that the elements in the vector $x(k)\ \ where\ \ k = 0,1,.....,n-1$ go on shifting upwards and finally we get the vector,

$$x(n-1)=\begin{bmatrix} d_{n-1} \\ 0 \\ | \\ | \\ 0 \\ 0 \end{bmatrix} \qquad (19)$$

and

$$x(n)=\begin{bmatrix} 0 \\ 0 \\ | \\ | \\ 0 \\ 0 \end{bmatrix} \qquad (20)$$

which assures that the terminal state i.e., the state at the n^{th} instant of sampling has been steered to the origin.

In fact, taking any values for $x(0)$ and solving for $x(1),\ x(2),.....,\ x(n)$ will continue to display the same nature of $x(k),\ k = 0,1,....,n$ as described above.

However, with arbitrary values of $d_i\ \ i=1,2,.....,n$ we require to use the modulo (2) operation on every element of the vectors $x(0),\ x(1),.....,\ x(n)$ in order to bring them in the binary form.

In order to avoid using modulo (2) operation, we can assume all the values of x (0) to be given by unity.

For example,

$$\begin{aligned} &x(0) &&= [1\ \ 1\ \ -\ \ -\ \ -\ \ -\ \ -\ \ 1]^T \\ &then && \\ &x(1) &&= [1\ \ 1\ \ -\ \ -\ \ -\ \ -\ \ 1\ \ 0]^T \\ &| && \\ &| && \\ &x(n-1) &&= [1\ \ 0\ \ -\ \ -\ \ -\ \ -\ \ -\ \ 0]^T \\ &x(n) &&= [0\ \ 0\ \ -\ \ -\ \ -\ \ -\ \ -\ \ 0]^T \end{aligned} \qquad (21)$$

3. Design of the Code

From (21) it can be observed that all the vectors, $x(0),\ x(1),.....,\ x(n-1)$ are linearly independent and hence let us consider them to form a generating set of vectors to form the code C, such that

$$G\ [I \quad P]$$

where P is the set of all the above linearly independent vector $x(0),\ x(1),.....,\ x(n-1)$, and I is correspondingly sized unit matrix.

Thus

$$G=\begin{bmatrix} 1 & 0 & 0 & - & - & 0 & | d_1 & d_2 & - & - & - & d_k \\ 0 & 1 & 0 & - & - & 0 & | d_2 & d_3 & - & - & d_k & 0 \\ & & & | & & & | & & & | & & \\ & & & | & & & | & & & | & & \\ 0 & 0 & - & - & 0 & 1 & | 0 & 0 & - & - & - & 0 \end{bmatrix}$$

Since this is linear code, we obtain all the code words by taking the sum of all the rows with each other and finally appending a zero vector to this set of code words so that this set of code words forms a sub-space.

This, in general, gives rise to the linear (n,M,d) code given below :

$$\begin{aligned} &\{0\ \ 0\ \ 0\ \ -\ \ -\ \ -\ \ -\ \ 0, \\ &\ 1\ \ 0\ \ 0\ \ -\ \ -\ \ d_1\ \ d_2\ \ -\ \ -\ \ d_k, \\ &0\ \ 1\ \ 0\ \ -\ \ -\ \ d_2\ \ d_3\ \ -\ \ d_{k-1}\ \ 0, \\ &-\ -\ -\ -\ -\ -\ -\ -\ -\ -\ -, \\ &0\ \ 0\ \ -\ \ -\ \ 0\ \ 1\ \ 1\ \ 0\ \ -\ \ 0, \\ &1\ \ 1\ \ 0\ \ -\ \ -\ \ 0\ \ d_1+d_2\ \ d_2+d_3\ \ -\ \ -\ \ d_{k-1}+d_k\ \ d_k, \\ &1\ \ 1\ \ 1\ \ 0\ \ -\ \ -\ \ 0\ \ d_1+d_3\ \ d_2+d_4\ \ -\ \ -\ \ d_{k-2}+d_k\ \ d_{k-1}\ \ d_k, \\ &-\ -\ -\ -\ -\ -, \\ &0\ \ 0\ \ -\ \ -\ \ 1\ \ 1\ \ d_k+1\ \ 0\ \ -\ \ -\ \ -0\} \end{aligned}$$

In the next section, we state and prove the main result of the paper, we establish that the total number of code words will be q^k, with $d_{\min}=2$.

4. Main Result

Before we state the main result of the paper, we reproduce the following Lemma from [3] which is crucial in the proof of the main result.

Lemma 1:[3] *For a linear code, the minimum distance d^* satisfies*

$$d^* = \min_{c \neq 0} w(c) = w^*$$

where the minimum is over all code words except the all-zero codeword.

***Proof*:**

$$d^* = \min_{\substack{c_i, c_j \in \mathbb{C} \\ i \neq j}} d(c_i, c_j)$$

$$d^* = \min_{\substack{c_i, c_j \in \mathbb{C} \\ i \neq j}} d(0, \ c_i - c_j)$$

$$d^* = \min_{\substack{c \in \mathbb{C} \\ c \neq 0}} w(c)$$

$$d^* = w^* \qquad (7)$$

Theorem 1: Given a p^{th} order discrete time completely controllable linear system

$$x(k+1) = Ax(k) + bu(k)$$

there exists an (n, M, d) code over $\mathbb{F}_q$ with q = 2 such that

1. $n = 2p$
2. $M = 1 + p + \left[{}^pC_2 + {}^pC_3 + ... + {}^pC_p \right]$
3. $d_{\min} = 2$

***Proof*:**

Note that the set of code words is generated by the basis vectors $x(0), x(1), ..., x(n-1), x(n)$ which are obtained by solving the given linear discrete-time controllable system, hence it is a vector space.

1. This is obvious from the size of G.
2. Since this is a linear Code, total number of code words shall be given by
   M = (all zero code) + (all the p code vectors) + { [Sum of all the code vectors taken two at a time] + [Sum of all the code vectors taken three at a time] + ……. + [Sum of all the code vectors taken (p-1) at a time] +[Sum of all the p code vectors] }
   $$M = 1 + p + \left[{}^pC_2 + {}^pC_3 + ... + {}^pC_p \right]$$
   $$M = q^p$$
3. This is obvious from the proof of Lemma 1.

The set of codewords so designed exhibit, all the properties which are stated in the main result and summarized below :

Order of the system (p)	Total no. of Codewords (q^p)	Size of the Generator Matrix	Length of the Codewords ($2p$)
2	4	4 x 2	4
3	8	6 x 3	6
4	16	8 x 4	8
5	32	10 x 5	10

Table 1 : Summary of the Main Result

5. Decoding

The existing conventional methods of coset leaders are applicable to decode the code which we have been able to design. However, since we have proved that $d_{\min} = 2$ for the codes designed in this paper, we propose a decoding algorithm which successfully performs error correcting upto 1-bit.

An algorithm devised by us for Decoding the proposed code

1. Consider our code consists of m code words C_i, each of length n
2. For each code word C_i we generate a table consisting of n words which are at a distance of 1 from C_i by changing each of the n symbols at a time.
3. Then via bit by bit comparison we eliminate those C_i's whose set of words at a distance 1 are the same or some of the words within their sets are same
4. Thus finally the total number of code words C_i within our code will be reduced, but then we are guaranteed of perfect decoding by our technique
5. Whenever a code word is received (say R_i), we look up the word in each of these m tables via bit by bit comparison
6. If the received code word is found in a table of particular code word C_i, we conclude that the receive word R_i maps onto that particular code word C_i.

5. Conclusion

The design of a binary code proposed here is based on the definition of controllability of discrete-time system. It has been proved that it is an (n, M, d) code with, $M = 2^p$ and $d_{\min} = 2$. It has also been observed that, this code satisfies the Plotkin as well as the Singleton upper bounds. The methodology for the generation of the code is very simple. For the generation of the code it is sufficient to simply know the vector formed by a set of initial conditions and the parity check matrix will automatically be formed.

The set of codewords so designed exhibit, all the properties which are stated in the main result. The existing conventional decoding scheme of coset leaders is applicable to decode the code proposed here. Further to this, for the codes designed in this paper, we have also devised an algorithm to propose a methodology to decode this code, with an error-correcting capability of upto 1-bit, its minimum distance being $d_{\min} = 2$.

References

[1] Kalman R. E., 'On the General Theory of Optimal Control', Proc. First Intl. Congr. of the Fed. of Automat Contr., (IFAC) Moscow, Vol. 1, pp. 481-493, (1960)

[2] Agashe S. D., Lande B. K., 'A New Approach to State-transfer Problem, J. Franklin Inst., Vol. 333(B), No. 1, pp. 15-21, (1996)

[3] Blahut R. E., 'Theory and Practice of Error Control Codes', Addison-Wesley Publishing Company, pp. 46-47, (1984)