ORIGINAL ARTICLE

Pseudo Random Number Coding for Lift Car Position Sensing

Mingxuan Du*

Shanghai Qibao Dwight High School, Shanghai 200135, China. E-mail: michelle-04@qq.com

Abstract: Aiming at the disadvantages of the traditional positioning for lift car applications, a novel binary bar coding and sensing technology is presented in this paper. It studies the relationship of the addressable length and the number of digits to be sensed. A more systematic approach is demonstrated to generate the pseudo random code more efficiently than the random sorting method used in the example case. Finally, a function of vector distance is developed to measure the difference of any two groups of the digits to be sensed along the complete code sequence. By searching the maximum of the vector distance, the optimal sequence can be achieved to reduce the chance of decode error.

Keywords: Lift Car Positioning; Position Addressability; Pseudo Random Code; Vector Distance

1. Introduction

When entering a lift with a trolley, the trolley might be stuck at the door. Sometimes people might have feelings of not walking on a flat ground when people enters the lift car. This is due to the lack of accuracy in the positioning system of lift cars. The problem that the car door opens and closes at the improper position causes more serious incidents in the past years. For instance, it was reported that on June 3, 2006, a male high school student was suffocated to death in an elevator of an apartment complex in Tokyo's Minato Ward when he was caught between the floor of the elevator and ceiling of the 12th floor[1]. See **Figure 1.**

The standard practice of the lift car positioning is mounting switch sensors at the every floor. When the car reaches the floor, the switch sensor will output a pulse signal to the lift controller. However, due to the inaccuracy of installation or the communication problem, the control may interpret a wrong position of the lift car from the sensor output signal. Research of how lifts determine where their cars more accurately has been conducted. Honeywell uses an array of magnetoresistive sensors which involve sensing a magnet attached to the car to offer more position information when the car approaches each floor. Its disadvantage is that magnetism wears off[2]. See **Figure 2.**

One-dimension bar code system has been introduced as a breakthrough positioning technology for logistic equipment. However, according to the work presented by Zhang and Li[3], the image processed to recognize the bar code and to tell the position is rather complicated. Consequently, the technology is not cost-effective. See **Figure 3.**

Figure 1. Demonstration of the Minato Ward Elevator Incident.

doi: 10.18686/esta.v7i3.142

Figure 2. Magnetoresistive sensors made by Honeywell.

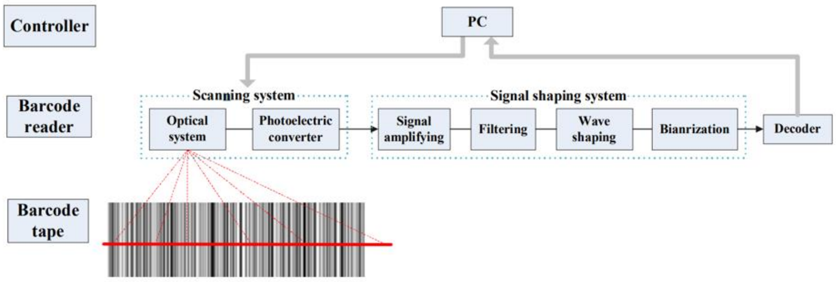

Figure 3. The schematic diagram of Zhang and Li's bar code positioning system.

1	0	1	0	0	1	0	1	1	1

Figure 4. Representation of the new bar code system which gives the value of 1010010111.

2. Background information

This article proposes a novel bar code method which can absolutely address different positions in a long distance. Unlike normal cases of bar codes that use black blocks with different widths to represent 1~9, the new coding is a binary system consisting of 1 and 0 only with unit width. For instance, a unit black is used to represent 1 while a unit white represents 0. See **Figure 4**.

Each tile of the bar code determines the positioning resolution. In the lift application case, it can be made as 1 millimetre so that even if the scanning becomes inaccurate, a deviation of 1mm would be hard for humans to identify and recognize. For many apartment buildings, there are 30 storeys in approximately 90m, which means the coding scale consists of at least 90,000 units of the digits.

In general, a function $H = 2^n$ can be applied, where H is the height of the building, and n is an integer that represents the number of digits the sensor needs to scan. Inversely, the number of digits to be sensed can be calculated if the H is given.

$n = \log_2(H)$

In the case of 16mm, the sensor needs to scan 4 digits at a time.

3. Procedures to generate the code

(1) First, convert 0~15 into binary form so that all the binary of four digits are shown. See **Table 1.**

Decimal	Binary
0	0000
1	0001
2	0010
3	0011
4	0100

Table 1 (continued).

Decimal	Binary
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Table 1. Converting 0~15 into binary

In order to form a 16-digit binary, only 4 of them are needed. As a result, there should be 16P4 = 43,680 possibilities.

(2) However, as the 4-digit scanning is continuous, lots of cases will face coding issue. For example:

0000000100100011

This is how the binary would be presented when keeping it in the same order as the decimal one. But, when the sensor scans the first position, from digit 1 to 4, it receives 0000. Moving on to the second position, from digit 2 to 5, it receives 0000 again. Then the processing unit would have trouble understanding in which position the car actually is.

(3) Several sequences of digits make every position a different 4-digit binary.

0010100110111000
1101011001000111
0010100110111100
1101011001000011
0101001101110000
1010110010001111

This code is not enough for locating the position. A look-up table is also required. For the case of 0010100110111000, the look-up table would be as follows.

Position	Binary code
1	0010
2	0101
3	1010
4	0100
5	1001
6	0011
7	0110
8	1101
9	1011
10	0111
11	1110
12	1100
13	1000

Table 2. Look-up table for the binary sequence 0010100110111000

(4) After finishing these sequences, it is true that three sequences of the original 4-digit binary are not being used because compared to only adding one digit to form the following positions, determining the first position requires four digits, or three extra digits. That is to say, alteration to the H (n) function is to be made, which becomes H (n) = $2^n + n - 1$.

4. Conclusion

After this investigation, generating the sequence is still a random process. This means for a real-life scenario like the one described in the above, ensuring that a function can be applied is vital. More researches are conducted about the binary system along with the positioning system. According to Petriu, this can be accomplished by applying a function of $R(0) = R(a) \oplus c(a-1)\times R(a-1) \oplus ... \oplus c(1)\times R(1)$. Here, R is identified as the value of the a^{th} position. ⊕ is the symbol that means XOR in Boolean operation which illustrates the difference in the value of the two results in a 1, while the two being the same results in a 0. c is a constant that can be either 0 or 1[4]. For instance, the sequence in the previous section 1101011001000011 has a 0 in its 4^{th} slot and a 1 in its 1^{st} slot, and then having a 1 as its zero slot, or last slot, is valid. See **Figure 5**.

But this is not the end. Linking the binary value to its corresponding position by inversing the previous function, it is $R(n+1) = R(1) \oplus b(2) \times R(2) \oplus ... \oplus b(n)\times R(n)$. Again, b is a constant. For example, in the sequence of 0101001101110000, the 5^{th} position, 0111, needs to shift for 8 intermediate states to reach the first position 0101.

0101001101110000

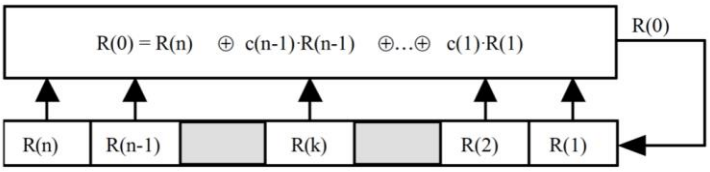

Figure 5. Procedure of generating such sequence by Petriu (n is adapted to a).

Shift register length n	Feedback for direct PRBS $R(0)=R(n) \oplus c(n-1)\cdot R(n-1) \oplus ... \oplus c(1)\cdot R(1)$	Feedback for reverse PRBS $R(n+1)=R(1) \oplus b(2)\cdot R(2) \oplus ... \oplus b(n)\cdot R(n)$
4	$R(0)=R(4) \oplus R(1)$	$R(5)=R(1) \oplus R(2)$
5	$R(0)=R(5) \oplus R(2)$	$R(6)=R(1) \oplus R(3)$
6	$R(0)=R(6) \oplus R(1)$	$R(7)=R(1) \oplus R(2)$
7	$R(0)=R(7) \oplus R(3)$	$R(8)=R(1) \oplus R(4)$
8	$R(0)=R(8) \oplus R(4) \oplus R(3) \oplus R(2)$	$R(9)=R(1) \oplus R(3) \oplus R(4) \oplus R(5)$
9	$R(0)=R(9) \oplus R(4)$	$R(10)=R(1) \oplus R(5)$
10	$R(0)=R(10) \oplus R(3)$	$R(11)=R(1) \oplus R(4)$

Table 3. The functions concluded by Petriu

5. Reflection

The number of possibilities of generating a 16-digit binary was previously hypothesized to be 2^{16}, which is 65,536. After obtaining the decimal-binary table and coming up with the idea of 16P4, the difference of 21,856 points out that using 2 powered by the number of digits has even more repeating results than 16P4. Thereby, the common function for the number of possibilities should be N = 2^nPn, where N is defined as the number of possibilities.

Comparing to the process in this investigation and Petriu's train of thought, using this method would end up with a huge look-up table as the number of digits increases, which is a huge waste for the memory of the control unit of lifts.

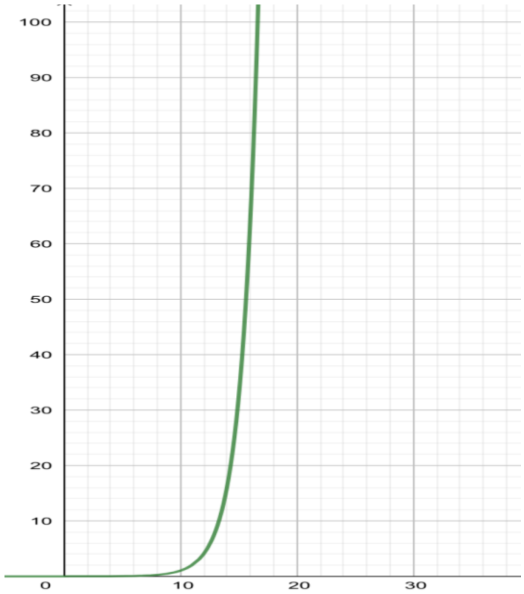

Figure 6. Graph of H (m) = $2^n + n - 1$.

The number of sensors	Heights of buildings (m)
13	8.20 ~ 16.3
14	16.4 ~ 32.7
15	32.8 ~ 65.5
16	65.6 ~ 131
17	132 ~ 262
18	263 ~ 524
19	525 ~ 1048
20	1048 ~ 2097

Table 4. Heights of buildings (m) vs. the number of sensors

Additionally, the function of $H = 2^n + n - 1$ is above the capacity to come up with an inverse function. Therefore, drawing a graph of the function is an alternative tool to determine the number of digits that the sensor needs to scan from the height of the building. See **Figure 6.**

On the other hand, H is so various that having multiple models with different numbers of sensors is too costly and useless. Therefore, the relation between the range of heights of buildings and the number of sensors is shown in **Table 4**.

As buildings lower than 8m hardly require a lift, and buildings with a height of more than 2097m are rare, the table only focuses on cases in which the number of sensors are between 13 and 20.

Just like other types of sensors related to optics, there will always be cases in which flying objects or stains on the bar code interfere with the signal, therefore affecting the result. Here, a function is developed that summarizes the distance in any two readings of a sequence to help improve error-tolerance. Given that D represents the vector distance of the n digits of the sequence, $x_1, x_2, x_3, \ldots, x_n, \ldots, x_M$.

$$D1 = \sum_{i=1}^{n} (x1, x2, x3, \ldots, xn) \oplus \begin{pmatrix} x2 + i \\ x3 + i \\ x4 + i \\ \ldots \\ xn + 1 + i \end{pmatrix}$$

$$D2 = \sum_{i=1}^{M-n} (x2, x3, x4, \ldots, xn + 1) \oplus \begin{pmatrix} x3 + i \\ x4 + i \\ x5 + i \\ \ldots \\ xn + 1 + i \end{pmatrix}$$

…

$$Dm = \sum_{i=1}^{M-n-m} (xm, xm + 1, xm + 2, \ldots, xm + n) \oplus \begin{pmatrix} xm + i \\ xm + ! + i \\ xm + 2 + i \\ \ldots \\ xn + m + i \end{pmatrix}$$

Summing up the total value of D, there goes:

$$D = \sum_{k=1}^{N} Dk$$

By processing the maximum value of D, the optimal sequence of binary is achieved.

References

1. Takuya N. Schindler elevator accident [Internet]. Available from: http://www.shippai.org/eshippai/html/index.php?name=nenkan2006_04_SchindlerE.
2. Honeywell. Sensors and switches in commercial traction elevators [Internet]. Available from: https://sensing.honeywell.com/honeywell-sensors-switches-commercial-traction-elevators-000695-2-en.pdf.
3. Zhang W, Li D. Research on barcode image binarization in barcode positioning system. International Journal of Computer Science Issues 2012; 9(5): 108–112.
4. Petriu EM. Absolute position measurement using pseudo-random binary encoding. Instrumentation & Measurement Magazine IEEE 1998; 1(3): 19–23.