Conference Applications of Mathematics 2013
in honor of the 70th birthday of Karel Segeth.
Jan Brandts, Sergey Korotov, Michal Křížek,
Jakub Šístek, and Tomáš Vejchodský (Eds.),
Institute of Mathematics AS CR, Prague 2013

DETECTION CODES IN RAILWAY INTERLOCKING SYSTEMS

Lucie Kárná[1], Štěpán Klapka[2]

[1] Faculty of Transportation Sciences, Czech Technical University
Na Florenci 25, Praha 1, Czech Republic
karna@fd.cvut.cz

[2] AŽD Praha s.r.o., Research and Development
Žirovnická 2, Praha 10, Czech Republic
klapka.stepan@azd.cz

Abstract

This paper describes a model of influence of random errors on the safety of the communication. The role of the communication in railway safety is specified.

To ensure a safe communication, using of safety code is important. The most important parameter of the safety code is the maximal value of the probability of undetected error. Problems related with computing of this value are outlined in the article. As a model for the information transmission the binary symmetrical channel is introduced.

The usability of the concept of a 'proper' code is discussed.

1. Introduction

This article discuss safety of communication between components of a railway interlocking systems (for example level crossings). The term *safety* is defined as absence of unacceptable level of hazard. This definition is not quite understandable without an additional explanation. However, for purpose of this paper it is sufficient to consider the word “safety” in its common sense.

The safety of a system has two main aspects. A *functional safety* concerns the manner how the system reacts on various combinations of outer inputs and its inner states (“what does it do?”). A *safety integrity* means ability of the system to really perform required functions (“does it really work?”). The safety integrity concerns the resistance of the system against both systematic and random errors. Nevertheless, only the requirements on the integrity in relation to random errors can be quantified.

This paper focuses on only a small part of the safety issues, in particular on a model of influence of random errors on the safety integrity, namely on communication safety.

1.1. Communication safety

Let us introduce the term "safe communication". A safe communication must ensure the following requirements:

- a message originates from the intended source (message *authenticity*),
- received information is complete and unchanged (*integrity*),
- messages are delivered in the right time (*timeliness*), and
- in the right sequence (correct *ordering*).

Some applications require *confidentiality* as an additional safety service – that the information cannot be disclosed to unauthorized subjects.

Many techniques can be used to ensure the *safety services* introduced above. Since every from these techniques provides protection against separate elementary errors, usually combination of several of them is employed. We can add a sequence number, a time stamp, or source and recipient identifier to the message. We can check the maximum time delay between two messages. The receiver can send an acknowledge message back to the sender. We can introduce a more sophisticated procedure of identification of communication participants. We can secure the message by a safety code or by cryptographic techniques.

The safety code has a special position among defense techniques, as it is the unique method of protection of messages against corruption. A safety-responsible protocol layer then should implement safety code to ensure integrity of messages. International safety standards for various types of systems state the usage of safety codes as mandatory requirement (for example [1] for railway applications).

2. Safety codes

An *error detection code* is a code detecting presence of some amount of errors in received messages. Error detection codes are used to overcome or reduce the impact of communication channel errors. However, these codes cannot provide a perfect protection and some amount of residual errors passes through undetected. Quantification of probability of occurrence of a residual error is a keystone of the probabilistic safety integrity study. A *safety code* is an error detection code used as a means to ensure safety in safety relevant communication system.

2.1. Linear binary codes

The "code-related" terminology in this paper is based on terms used in mathematical coding theory (see for example [3]). In this article we restrict ourselves only to linear binary detection codes, with codewords of length n bits, and with k information bits, defined as follows:

A linear binary $(n,k)-$*code* $\mathbf{K}$ is any k-dimensional subspace of the space $\mathbf{Z}_2^n$. Traditionally, binary vectors from $\mathbf{Z}_2^n$ are called *words*; the words from the code $\mathbf{K}$

are *codewords.* In the $(n, k)-$code the codeword length is n, number of information bits is equal to k and number of redundant bits is equal to $n - k$.

The most simple example of the linear binary code is a parity check. The *even parity* code consists from all words of the given length n, in which the count of ones is even. This code has $n - 1$ information bits and 1 redundant bit. The parity check is used as a safety code in most harware and software applications.

2.2. Error detection

In transfer of the encoded information in the space (transmission and reception of the message) or in the time (usage of data storage medium to record and later restore the message) the message can be modified by various external influences. On the level of individual bits, a modification can manifest by missing or superfluous bit(s), or by altered bits with overall number of bits preserved. In this paper, we ignore the first type of modification (synchronization slip) and focus solely on the second type – modifications that do not change the number of bits.

Let us describe the mechanism of detecting these modifications. A source intends to send a k-bit message. The error detecting code generates an n-bit codeword u, and the source transfers this codeword. A target receives an n-bit word v from $\mathbf{Z}_2^n$, not necessarily a codeword. If the received word v is not a codeword, then the receiver detects an *error.*

The second possibility is that the received word v is a codeword. Then there are two possible scenarios: The received codeword v is equal to the original codeword u, because there were no modifications in transfer. Alternatively the received codeword v is different from the original codeword u, because a modification during transfer unfortunately creates some codeword. The receiver has no possibility to recognize, which one from this scenarios occurs. The second scenario is then bad and results in an *undetected error.* The probability of such undetected error of error detection codes used in safety relevant applications (including transportation control) is very important safety parameter.

The difference $v - u$ between the received word v and the original word u is called an *error word.* The undetected error words of a linear code are all nonzero codewords of the given code, due to its linearity (see for example [3]). This is a great advantage of using of linear codes, as this make probability calculations more feasible than for other types of codes.

2.3. Weight structure

We define the *Hamming weight* of a word as the count of non-zero bits in the word. Then we define the *minimal distance* of a linear code as the smallest non-zero Hamming weight of its codeword.

The minimal distance of a linear code sets the ability of the code to detect some classes of transmission errors. A code with minimal distance d will detect all errors with at most $d - 1$ modified bits in transmitted codeword (see [3]). Such a code will not detect all errors with d or more modified bits. Nevertheless, some cases of modifications of d or more symbols will be detected. Various codes with equal n, k, and d differ in their capability of detecting modifications with d or more changes, and thus such codes differ in their undetected error probability.

For more detailed description of the code we define a *weight structure* of the code as a vector $A = (A_1, A_2, \ldots, A_n)$, where A_i denotes count of codewords with Hamming weight equal to i. For linear codes, the weight structure is fully sufficient for description of the ability of the code to detect modifications of d or more symbols, as we show in the following analysis.

2.4. Probability of undetected error

Let us derive a formula for the probability of undetected error of binary linear code. This probability is equal to probability of receiving a non-zero codeword if the zero codeword is transmitted (for details see [3]).

Consider a transmission of the zero codeword. Suppose we received a word with exactly i non-zero bits. The probability that the received word is a codeword is the ratio between the count of all codewords with i non-zero symbols (A_i from the weight structure of the code), and count of all words with i non-zero bits $\binom{n}{i}$. Denoting P_i the probability that received word has exactly i wrong bits, then the probability P_{ud} of an undetected error of the code is equal to

$$P_{ud} = \sum_{i=1}^{n} P_i \frac{A_i}{\binom{n}{i}}. \tag{1}$$

The probability P_i that exactly i bits are modified during transmission is independent of the code properties, and depends solely on conditions of the information transmission.

There are various models of communication channels, with varying characteristics. Choosing the right model of a communication channel that corresponds to real-world conditions, and produces useful results in our calculations is rather a difficult process.

2.5. Binary symmetrical channel

The most frequently used transmission channel model is a *memoryless binary symmetrical channel (BSC)*. This is a simple probability model based on a bit transmission, parametrized by a constant probability of bit modification p_e (*bit error rate*). In this model, a transmitted bit is modified during the transmission with the probability p_e, regardless of its original value and independently of other bits in the transmission.

In the BSC model, the probability that a word with n bits is received with i bits modified is equal to

$$P_i = \binom{n}{i} p_e^i (1 - p_e)^{n-i}. \tag{2}$$

Substituting (2) to the formula (1), for the probability of undetected error we get:

$$P_{ud}(p_e) = \sum_{i=1}^{n} {p_e}^i (1 - p_e)^{n-i} A_i. \qquad (3)$$

For a final calculation, it is necessary to know A_i, the quantities of codewords with the Hamming weight equal to i. A binary linear (n, k)-code is a k-dimensional linear subspace of $\mathbf{Z}_2^n$. It has exactly 2^k elements, and thus

$$A_0 + A_1 + \ldots + A_n = 2^k.$$

As every linear code contains a zero word, A_0 equals one. For minimal distance of the code equals to d, A_1, A_2, ..., A_{d-1} equals zero.

Other values of A_i are known only for a few types of specially constructed codes. The calculation of A_i for a general type of a code takes a lot of time and involves generation of 2^{n-k} codewords. (For more details see for example [3].) The generation of the codewords is not complicated and easily parallelizable, however the time spent is still enormous even for commonly used values $n - k$ (32, 48, 64 or 96).

2.6. 'Good' and 'proper' code

Because computing of the weight structure of a code is very troublesome, there are efforts to find some more manageable method of determination of the probability of undetected error of the code.

First, it is not necessary to know a complete course of the function $P_{ud}(p_e)$; for subsequent safety considerations its maximum value is sufficient.

Second, we need not consider all possible values of the bit error probability p_e. A channel with bit error probability $p_e = 1$ exactly inverts every transmitted message. Generally, channels with bit error probability higher than $1/2$ have tendency to invert messages rather than transmit them unchanged. If a code does not contain a word with all bits equal to one, then all inversions of a codeword are detected. Therefore, for such codes it is sufficient to consider values of the p_e in the interval $[0, 1/2]$.

Introducing the boundary value $p_e = 1/2$ into formula (3) for undetected error probability in the BSC, it follows

$$P_{ud}(1/2) = \frac{2^k - 1}{2^n} < 2^{k-n}.$$

We underline that this estimate is the same for every binary linear $(n, k)-$code. Moreover, the estimate $P_{ud}(p_e) < 2^{k-n}$ it fulfilled in some small neighbourhood of $1/2$, but not on the whole interval from zero to one.

In the case that the estimate $P_{ud}(p_e) < 2^{k-n}$ is valid on the whole interval $[0, 1/2]$, we need not examine the code any more. This is a motivation for following definitions of the terms 'good' and 'proper' code:

- A binary linear (n, k)-code is *'good'*, if for all $p_e \in [0, 1/2]$ the inequality $P_{ud}(p_e) < 2^{k-n}$ is valid.
- A binary linear (n, k)-code is *'proper'*, if the function $P_{ud}(p_e)$ is monotone increasing on the interval $[0, 1/2]$.

It is evident that a 'proper' code is always 'good' as well, and after then the term 'proper' seems to be redundant. However, this term has its sense: on less erroneous channels the 'proper' code has a lower failure probability than on more erroneous ones. This is reasonable behaviour.

The second reason for introducing the term 'proper' code is that the monotonicity of the function $P_{ud}(p_e)$ can be generally proven for some classes of codes. These codes are consequently 'good' and theirs probability of undetected error in the BSC is upperbounded by the known value 2^{k-n}, which can be used in following safety calculations of the whole system.

As consequence of this, it was widely supposed among safety engineers, that all "reasonable" codes are 'proper', or at least "almost proper". In fact, codes really used in practice very often are not 'proper' and their probability of undetected error exceeds the value 2^{k-n}, often very strongly. Usually this occurs for relatively low values of the bit error rate p_e. For example, we found a code with maximal value of the probability of undetected error more than thousand times higher than 2^{k-n}. Therefore, the execution of a probabilistic analysis using BSC is necessary in all cases.

The evaluation of the maximal value of the probability of undetected error has to be done numerically. When the code is not 'proper', the most successful procedure is based on the Newton's method with adaptive precision computation. As the function $P_{ud}(p_e)$ is almost constant on the most part of its rank, this calculation is complicated and time-consuming as well. For recognition that the code is not 'proper', it is very useful to use the binomial moments (for more details see [2]).

3. Conclusion

The calculation of the maximal value of probability of undetected error is a laboured and often very lengthy process. Nevertheless, it is essential for evaluation safety parameters of the whole systems and cannot be omitted. On the other hand, safety codes contribute to the overall safety of the railway traffic only by a small part.

The reasons of most railway accidents are simple. In the past, unpredictable technical failures was frequent. Presently accidents caused by neglecting of maintenance or by failure of operator predominate. Both of them are human factor failures. As far as we know, no one railway accident caused by failure of safety code was recorded.

Recent interlocking systems pass to unified interoperable communication interfaces, usually designated for employment in open transmission systems (European systems ERTMS/ETCS, GSM-R, EURORADIO protocol). Safety codes in this systems use cryptographic techniques. These cannot be evaluated by the above mentioned method.

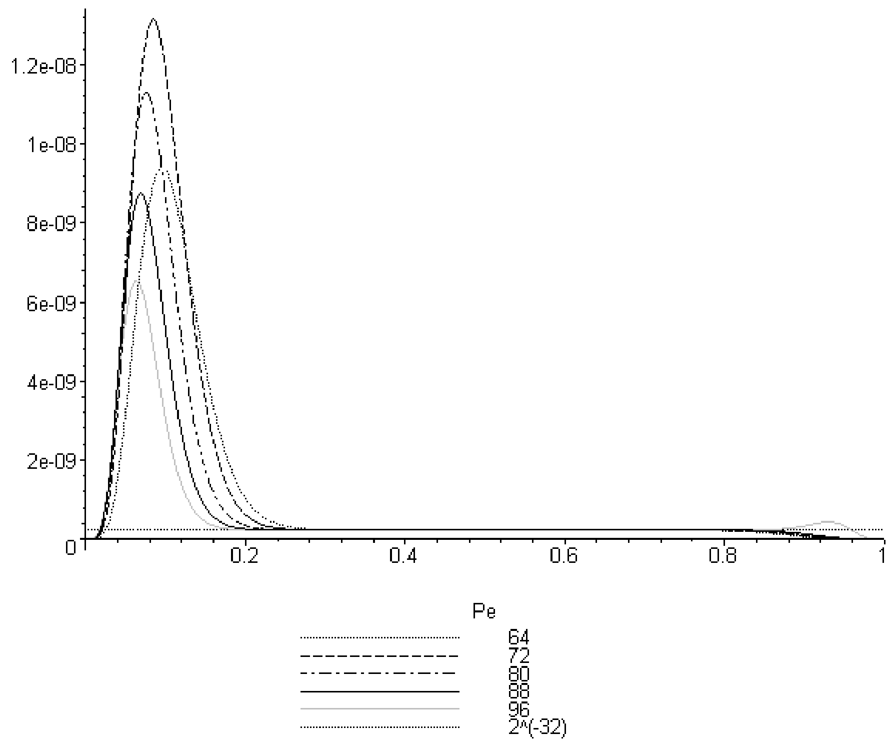

Figure 1: An example of codes, which are not 'proper'. These are five different codes, created by shortening of the same code. The codeword lengths n of these codes vary from 64 to 96 bits, number of redundant bits $k - n$ is 32 for all of them. The horizontal line near to the bottom edge of the graph is the constant 2^{-32}. The maximal value of probability of undetected error is more than 50-times higher than this value for the worst code with the codeword length 72 bits.

Another open problem is an ensuring of the independence between safety and transmission codes. Actually, there does not exist consensus even about the definition of this independence.

References

[1] EN 50159 Railway applications – Communication, signalling and processing systems – Safety-related communication in transmission systems, CENELEC, 2010.

[2] Dodunekova, R.: Extended binomial moments of a linear code and the undetected error probability. Probl. Peredachi Inf. **39**(3) (2003), 28–39. [Probl. Inf. Trans. (Engl. Transl.) **39**(3) (2003), 255–265].

[3] Huffman, W. C., Pless, V.: *Fundamentals of error-correcting codes.* Cambridge University Press, Cambridge, 2003, ISBN 0-521-78280-5.